BOF

Between

Ontologies and Folksonomies

Michigan State University-Mi, US
June 28, 2007

Workshop held in conjunction with

Preface

Today on-line communities, as well as individuals, produce a substantial amount of unstructured (and extemporaneous) content, arising from tacit and explicit knowledge sharing.
Various approaches, both in the managerial and computer science fields, are seeking ways to crystallize the - somewhat volatile, but often highly valuable - knowledge contained in communities "chatters". Among those approaches, the most relevants appear to be those aimed at developing and formalizing agreed-upon semantic representations of specific knowledge domains (i.e. domain ontologies).
Nonetheless, the intrinsic limits of technologies underpinning such approaches tend to push communities members towards the spontaneous adoption of less cumbersome tools, usually offered in the framework of the Web 2.0 (e.g. folksonomies, XML-based tagging, etc.), for sharing and retrieving knowledge.
Inside this landscape, community members should be able to access and browse community knowledge transparently and in a personalized way, through tools that should be at once device-independent and context- and user-dependent, in order to manage and classify content for heterogeneous interaction channels (wired/wireless network workstations, smart-phones, PDA, and pagers) and disparate situations (while driving, in a meeting, on campus).

The *BOF- Between Ontologies and Folksonomies* workshop, held in conjunction with the third *Communities and Technologies* conference in June 2007[1], aimed at the development of a common understanding of the frontier technologies for sharing knowledge in communities. We are proposing here a selection of conceptual considerations, technical issues and "real-life case studies" presented during the workshop. We believe that useful suggestions and guidelines for effective approaches to information sharing and retrieval can be envisioned starting from the high-valuable scientific works presented here.

The programme committee and organizers wish to thank all the authors who submitted papers, the moderators, the participants and everyone who has contributed to the success of the workshop.

October 2007

Dario Maggiorini
Alessandro Provetti
Laura Anna Ripamonti

[1] C&T 2007 : June 28-30, 2007, Michigan State University

STEERING COMMITEE

Dario Maggiorini	Dept. of Information and Communication (D.I.Co.), University of Milan – Italy
Alessandro Provetti	Dept. of Physics, University of Messina – Italy
Laura Anna Ripamonti	Dept. of Information and Communication (D.I.Co.), University of Milan – Italy

PROGRAM COMMITTEE

Nadjib Achir	University of Paris XIII – France
Marco Bettoni	Swiss Distance University of Applied Science – Swiss
Khaled Bussetta	University of Paris XIII – France
Fiorella De Cindio	D.I.Co., University of Milano – Italy
Marilena De Marsico	D.I., University of Roma "La Sapienza" – Italy
Aldo de Moor	CommunitySense – The Netherlands
Cheng-Yen Wang	The Kaohsiung Open University – Taiwan

ORGANIZING COMMITTEE

Ines Di Loreto	D.I.Co., University of Milano – Italy
Francesco Giudici	D.I.Co., University of Milano – Italy
Massimo Marchi	D.S.I., University of Milano – Italy
Cristian Peraboni	D.I.Co., University of Milano – Italy

Table of Contents

Automated Information Extraction from Web Sources: a Survey

Giacomo Fiumara

Dipartimento di Fisica, Università degli Studi di Messina,
Salita Sperone 31, I-98166 Messina, Italy
giacomo.fiumara@unime.it

Abstract. The Web contains an enormous quantity of information which is usually formatted for human users. This makes it difficult to extract relevant content from various sources. In the last few years some authors have addressed the problem to convert Web documents from unstructured or semi-structured format into structured and therefore machine-understandable format such as, for example, XML. In this paper we briefly survey some of the most promising and recently developed extraction tools.

1 Introduction

Although XML can be regarded as a *lingua franca* of the Web, nowadays almost all information available in Web sites is coded in form of HTML documents. This situation in unlikely to change in short or even medium term for at least two reasons: the simplicity and power of HTML authoring tools, together with a valuable inertia to change markup language. From the point of view of anyone interested in extracting information from Web sites, on the opposite, the difference between HTML and XML is evident. Although they are both derived from SGML, HTML was designed as a presentation-oriented language. On the contrary, XML has among its points of strength the separation between data and its human-oriented presentation, which allows data-centered applications to better handle large amounts of data. Another fundamental advantage of XML is the availability of powerful instruments for querying XML documents, namely XQuery/XPath[2], together with the increasing availability of native XML Databases [1], see for example eXist[3] and Monet[4]. Whereas [15] has surveyed the tools for information extraction in the Semantic Web, this survey would like to examine the state of the art of tools addressing the traditional Web. Even though the taxonomy proposed in [15] is largely adopted here, the emphasis is on what can be done in the context of existing, legacy Web sites. Community Web sites that have been serving their users for long time are a particular case in point. This brief

survey will focus in Section 2 on the main questions regarding wrappers and their automatic generation and then give an overview of systems in Section 3. Related work will be presented in Section 4. Conclusions and future work will be presented in Section 5.

2 Wrapping a Web page

Information extraction from Web sites is often performed using wrappers. A wrapper is a procedure that is designed to access HTML documents and export the relevant text to a structured format, normally XML. Wrappers consist of a series of rules and some code to apply those rules and, generally speaking, are specific to a source. According to [6, 16] a classification of Web wrappers can be made on the base of the kind of HTML pages that each wrapper is able to deal with. Three different types of Web pages can be distinguished:

- *unstructured pages*: also called free-text documents, unstructured pages are written in natural language. No structure can be found, and only information extraction (IE) techniques can be applied with a certain degree of confidence.
- *structured pages*: are normally obtained from a structured data source, e.g. a database, and data are published together with information on structure. The extraction of information is accomplished using simple techniques based on syntactic matching.
- *semi-structured pages*: are in an intermediate position between unstructured and structured pages, in that they do not conform to a description for the types of data published therein. These documents possess anyway a kind of structure, and extraction techniques are often based on the presence of special patterns, as HTML tags. The information that may be extracted from these documents is rather limited.

Besides the HTML page structure, effective wrappers consider also the structure of hyperlink as it may reveal relevant information. Depending on the type of Web search engine the following kinds of results can be obtained:

- *one-level one-page* result: one page contains all the item descriptions;
- *one-level multi-pages*: a series of pages linked one to another, all containing the item description;
- *two-level pages*: a chain of pages, each containing a shortened description of items, each linking to a detailed page.

3 Information Extraction Tools

In this section a brief overview of some information extraction tools will be given. The idea is to illustrate the main features of tools belonging to the

family of the so-called *HTML aware tools* (see [16, 15, 6] for the related taxonomy). Among the large number of information extraction tools we chose Lixto and Fetch as examples of powerful yet commercial semi-supervised wrapper generators, while RoadRunner is a prototype of fully automatic tools. Finally Dynamo will be described as an example of extraction tools which rely on the cooperation between the webmasters of the Web sites which publish information and the user willing to automate the extraction process.

3.1 LiXto

The *LiXto* project was started by Gottlob et al. at TUWIEN and is now developed and sold by the LiXto GMbh software house. *LiXto* [7, 8] is a method for visually extracting HTML/XML wrappers under the supervision of a human designer. *LiXto* allows a wrapper to interactively and visually define information extraction patterns on the base of visualized sample Web pages. These extraction patterns are collected into a hierarchical knowledge base that constitutes a declarative wrapper program. The extraction knowledge is internally represented in a Datalog-like programming language called *Elog* [9]. The typical user is not concerned with *Elog* as wrappers are build using visual and interactive primitives. Wrapper programs can be run over input Web documents by a module in charge of extraction which then translates the output in XML. The latter is done thanks to a *XML translation scheme* with the possibility to construct a Document Type Definition (DTD) which describes the characteristics of the output XML documents. Among the most interesting features is the ability to access Web data even if protected by means of a username/password authentication mechanism, if the user provides them. LiXto has also the possibility to follow links thus collecting information even if spread across several Web pages, the flexibility to output extracted structured information into several formats, namely XML, SQL records and XHTML newly produced Web pages. Finally, the extraction process can be scheduled in order to be repeated at fixed times.

3.2 Fetch Agent Platform

Fetch Agent Platform [21] is another example of commercial information extraction tool.It is based on two major components, the AgentBuilder which provides a visual environment that allows a user to construct web agents, and the AgentRunner which automatically performs the tasks specified by the agent, and produces structured data. The framework also provides a tool able to monitor Web target pages, specifying which data fields are to be checked. The extraction rules are based on landmarks (groups of consecutive tokens) that enable a software agent to locate the start and end of fields within a page. The extraction algorithm that learns these landmarks based on examples labeled by the user and uses the hierarchical structure of the page to constrain the learning problem.

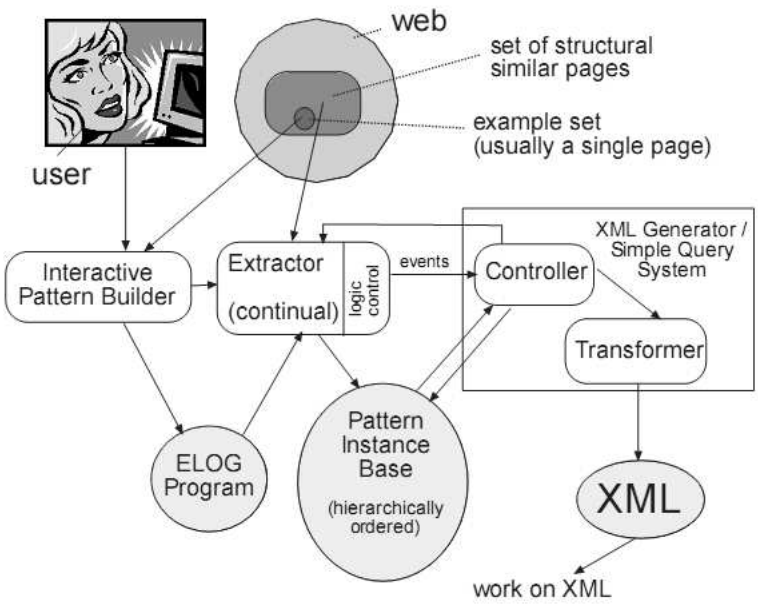

Fig. 1. The architecture of *LiXto*, from [7]

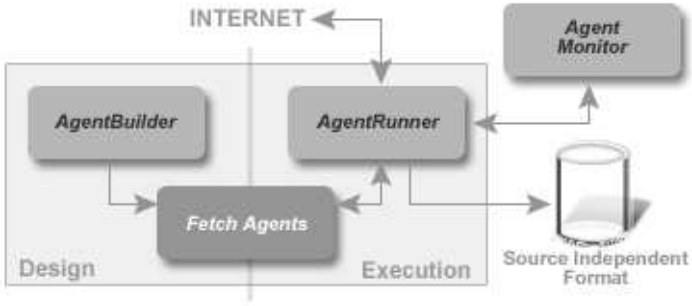

Fig. 2. The architecture of *Fetch Agent Platform*, from [21]

3.3 RoadRunner

RoadRunner [10, 11, 12, 13, 14] was developed at the University of Roma 3 and applies to intensive Web sites, i.e. those sites with large amounts of data and a rather regular structure. RoadRunner works by comparing the HTML structure of a set of sample pages of the same type, and generates a schema for the data contained in the pages. This schema is used as a starting point for the inference of a grammar which is capable to recognize the instances of

attributes identified for this schema in the set of sample pages. The extraction procedure is based on an algorithm that compares the tag structure of the set of sample pages and produces regular expressions able to handle structural differences found in the set of sample pages. A peculiar feature of RoadRunner is that this procedure is completely automatic and no user intervention is required.

3.4 Dynamo

The Dynamo Project [18, 19] addresses data extraction and channeling over legacy Web sites in plain HTML. Dynamo is intended to benefit two types of users. First, webmasters may employ it to manage the creation of RSS feeds, thus avoiding to do it by hand or by means of proprietary software. Second, users, i.e., consumers of feeds, may use it to overcome limitations such as i) old feeds may not be consulted and usually are deleted from servers and ii) traditional HTML servers cannot execute advanced queries directly. On the contrary, with Dynamo it becomes possible to:

- automatically and dynamically generate RSS feeds starting from HTML Web pages;
- store feeds in chronological order;
- query and aggregate them thanks to Web Services (WS) acting as agents.

It is important to stress that these results were obtained with a lightweight pull algorithm for retrieving HTML documents by Web servers, thus minimizing the required Web traffic for the updates of news sources [19].

HTML documents contain a mixture of information to be published, i.e., meaningful to humans, and of directives, in the form of tags, that are meaningful to the browsers and determine the appearance on the screen. Moreover, since the HTML format is designed for visualization purposes only, its tags do not allow sophisticated machine processing of the information contained therein.

Among other things, one factor that may prevent the spread of the Semantic Web is the complexity of extracting, from existing, heterogeneous HTML documents machine-readable information. Although the Dynamo project addresses only a fraction of the Semantic Web vision, management of HTML documents needs some technique to locate and extract some valuable and meaningful content. Therefore, a set of annotations, in form of meta-tags, were defined; they are inserted inside HTML in order to highlight informational content that is essential for the creation of a RSS feed. In this application, meta-tags are used as annotations, to describe and mark all interesting information, in order to help in the extraction and so-called XML-ization phases. Notice that with pages that are dynamically generated out of some template (which is the case with practically all on-line fora) Dynamo annotation is done, manually but only once and for all, over the page template.

Once HTML documents are processed by Dynamo, annotated semantic structures are extracted and organized into a simple XML format to be stored and used as a starting point for document querying and transformation. The structure of the XML output resembles the structure of meta-tags previously defined and the RSS XML structure, in order to facilitate transformations from the former to the latter. At the moment, a version of Dynamo is undergoing a phase of testing in several forum of the Milan Community Network (Rete Civica Milanese).

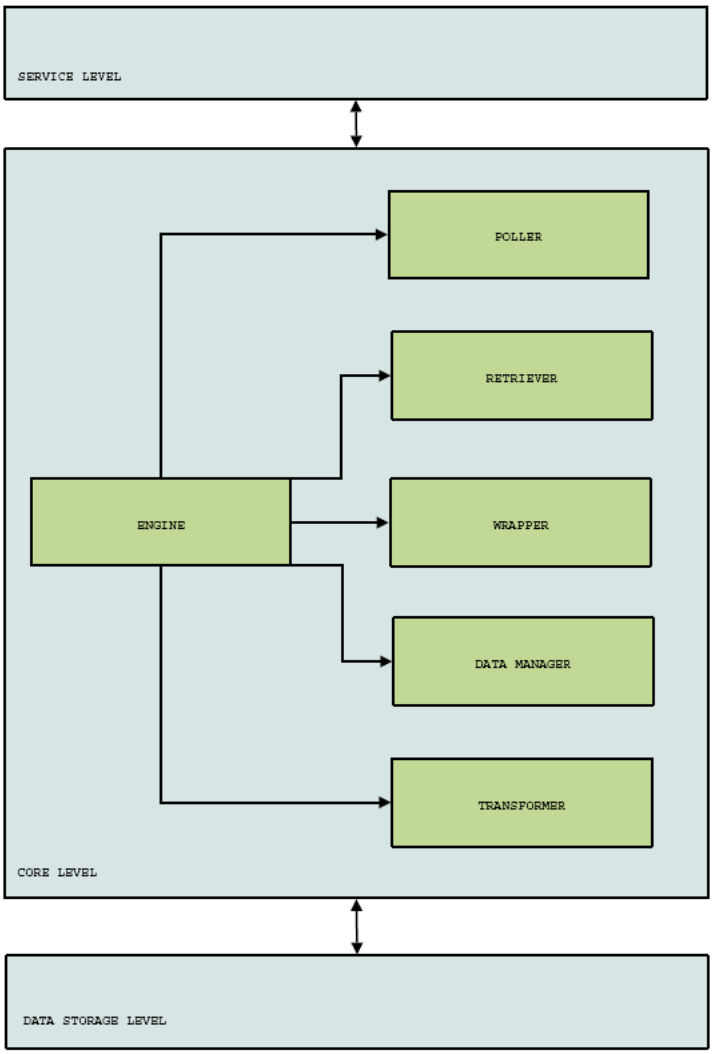

Fig. 3. The architecture of *Dynamo*

4 Related Work

In the past few years, many approaches to the problem of Information Extraction (IE) by means of Wrapper Induction (WI) systems have been tackled. Previously proposed taxonomies will be briefly examined in this section. Hsu and Dung [22] classified wrappers into 4 categories:

- hand-made wrappers using general-purpose programming languages;
- designed programming languages;
- heuristic-based wrappers;
- WI approaches.

A complete categorization was made by Laender et al. [15]. They proposed the following taxonomy:

- languages for wrapper development;
- HTML-aware tools;
- NLP-based tools;
- wrapper induction tools;
- modeling-based tools, and
- ontology-based tools.

They also compared among the tools using these features: degree of automation, support for complex objects, page contents, availability of a GUI, XML output, support for non-HTML sources, resilience and adaptiveness.

Sarawagi [24] classified Web sites wrappers according to the amplitude of the tasks they are able to face. So he distinguishes *record-level wrappers*, capable to extract elements of a single list from a Web page, *page-level wrappers* which extract elements of multiple records and, finally, *site-level wrappers* which can extract and convert into structured format an entire Web site.

More recently, Chang et al. [17] proposed a three-dimensional representation of IE features: the first dimension evaluates the difficulty of an IE task, the second compares the various techniques and the third dimension compares both the training effort of a user and the necessity to port an IE system across different domains.

5 Conclusions and future work

In this paper we presented a short survey of most recent tools for the extraction of information from Web sites. All the tools presented here automatically generate wrappers in order to accomplish their task and all of them provide output data in XML format, thus focusing on the meaning of data rather than on their graphical representation.

There are a series of current and future applications where information extraction tools can fully exploit their power. One of the most promising seem to be the comparison of items, for example in commercial aggregators.

The possibility for a user to compare different offerings of the same object is a feature currently not supported by online auction sites.

Even in the area of communication, the possibility of aggregating and querying information automatically extracted from different Web news sites seems really promising, specially in conjunction with the features offered by XML-based query engines. This, together with more flexible and powerful extraction tools will certainly help paving the road to the semantic web.

References

1. Bourret RP (2005) XML and Databases. http://rpbourret.com
2. W3C (2005) XQuery 1.0. http://w3c.org/TR/xquery
3. eXist (2007) Open Source XML Native Database. http://exist-db.org
4. MonetDB (2007) http://monetdb.cwi.nl
5. Muslea I. (1999) Extraction Patterns for Information Extraction Tasks: A Survey. American Association for Artificial Intelligence
6. Eikvil L. (1999) Information Extraction from World Wide Web - A Survey -. Technical Report 945, Norvegian Computing Center
7. Baumgartner R., Flesca S., Gottlob G. (2001) Visual Web Information Extraction with Lixto. In Proc. of VLDB, 2001
8. Baumgartner R., Flesca S., Gottlob G. (2002) Declarative Information Extraction, Web Crawling and Recursive Wrapping with Lixto. In Proc. of LPNMR, 2002
9. Baumgartner R., Flesca S., Gottlob G. (2002) The Elog Web Extraction Language.
10. Mecca G., Grumbach S. (1999) In search of the lost schema. ICDT(1999)
11. Crescenzi V., Mecca G., Merialdo P. (2001) RoadRunner: Towards Automatic Data Extraction from Large Web Sites. VLDB(2001)
12. Crescenzi V., Mecca G., Merialdo P. (2001) The RoadRunner Project: Towards Automatic Extraction of Web Data. ATEM (2001)
13. Crescenzi V., Mecca G., Merialdo P. (2001) Automatic Web Information Extraction in the RoadRunner System. DASWIS (2001)
14. Crescenzi V., Mecca G., Merialdo P. (2002) Wrapper Oriented Classification of Web Pages. ACM SAC (2002)
15. Laender A.H.F. , Ribeiro-Neto B.A., da Silva A.S., Teixeira J.S. (2002) A Brief Survey of Web Data Extraction Tools. SIGMOD Records 31(2) 2002
16. Flesca S., Manco G., Masciari E., Rende E. and Tagarelli A. (2004) Web wrapper induction: a brief survey. AI Communications 17 (2004) 57 - 61
17. Chia-Hui Chang, Kayed M., Girgis M.R., Shaalan K. (2006) A Survey of Web Information Extraction Systems. IEEE Transactions on Knowledge and Data Engineering, TKDE-0475-1104.R3
18. Bossa S. (2005) Gradation Project in Informatics. University of Messina (in Italian)
19. Bossa S., Fiumara G., Provetti A. (2006) A Lightweight Architecture for RSS Polling of Arbitrary Web sources. Proc. of WOA conference. Available from http://mag.dsi.unimi.it/

20. De Cindio F., Fiumara G., Marchi M., Provetti A., Ripamonti L.A. and Sonnante L. (2006) Aggregating information and enforcing awareness across communities with the Dynamo RSS feeds creation engine: preliminary report. OTM Workshops (1) 2006: 227-236
21. Fetch Technologies, available from http://www.fetch.com
22. Hsu C.-N. and Dung M. (1998) Generating finite-state transducers for semi-structured data extraction from the web. Journal of Information Systems 23(8): 521-538 (1998)
23. Chang C-H., Hsu C.-N. and Lui, S.-C. (2003) Automatic information extraction from semi-structured web pages by pattern discovery. Decision Support Systems Journal, 35(1): 129-147 (2003)
24. Sarawagi S. (2002) Automation in information extraction and integration, Tutorial of VLDB (2002)

(X)querying RSS/Atom Feeds Extracted from News Web Sites: a Cocoon-based Portal

Giacomo Fiumara, Mario La Rosa, and Tommaso Pimpo

Dipartimento di Fisica, Università degli Studi di Messina
Salita Sperone 31, I-98166 Messina, Italy
giacomo.fiumara@unime.it

Abstract. The Web is fastly becoming the predominant source for news and information for many people. In the past few years, a new delivery system has emerged in the form of RSS feeds. Such feeds normally provide a brief of a larger news posted on the Web. RSS feeds, collected to form "channels" according to some thematic criteria, can be accessed using Web browsers or specialized software called "news aggregators". Even so, the amount of information available on the Web still exceeds human possibilities. In order to allow more selective and precise user choice, we developed a Web Cocoon-based platform which selects and publishes news gathered from various news Web sites. The selection is done submitting XQuery queries to a local repository and exploits the intrinsically semantic nature of RSS feeds.

1 Introduction and motivation

The World Wide Web (the Web) has become the predominant source for news and information for many people. To address the vast amount of content and the high frequency of news publication, a new delivery system has emerged in the form of "channels" or "feeds." These feeds, which are supplied by Websites such as CNN and BBC News, can be read using traditional Web browsers or specialized software, called "news aggregators." The two main formats for these feeds are RSS [4](Really Simple Syndication or Rich Site Summary) and Atom [16, 6]. They both provide an XML-based summary of the informational content of a website, with a brief description of the new "article" and links to the actual content. Feeds provide easy access to content in a pro-active mode, but presenting users with more content that they can handle. Current news aggregators do not provide the users very efficient means, beyond a simple keyword search, by selecting the most relevant content. Over a span of time, users will repetitively consider and discard content that does not match their interests. One major point is the impossibility to query, even in the relational sense of the term, the feed repositories before the retrieving of the data sources. An obvious advantage of such a "remote" query, would result in

a reduction of network traffic, less computational efforts and more pertinent content. We present here the preliminary results of our project, consisting in a Web site which publishes feeds retrieved by means of a series of queries (in Xquery[23, 12, 10] language) submitted to feed repositories spread across the Web. Registered users have the possibility to propose new repositories to be included in the set of those to be queried.

2 The software platform

The instruments we used for the development of our project (the graphical interface of the portal has been written in XHTML[21], Ajax[17] and CSS[19]) are entirely based on XML technologies; the development platform is the Apache Cocoon framework[18, 3, 8, 11, 14], which well suits for the construction of web applications by means of the aforesaid technologies. In the following we describe in some detail both Cocoon framework, XQuery and its potential, XSP programming language[26].

2.1 Cocoon

Cocoon was born as a Java servlet [20] with the aim of transforming XML documents through XSLT stylesheets[24, 25]. The community which coalesced around this project led it to its actual form, that is a Web-publishing framework built on the concepts of SoC (Separation of Content) [31, 32, 33, 34, 35] and component-based development of Web applications. Cocoon realized that mission by the notion of pipeline of components, where each component carries out a specific operation. Its creators define it "the web glue for your web application development needs", because SoC allows different development phases to coexist, thus reducing the possibilities of conflicts and error propagation.

Cocoon is based on the Avalon model[27, 28] and inherits its best features: first of all, the possibility of defining and developing new components. Components are defined by a descriptive interface and an implementation. For example, a parser is described by a Java interface that specifies all services it has to guarantee. Since this parser must be used inside an application, it's necessary for implementation to be conforming to the interface.

Cocoon's most important innovation is SoC-based design. During Web development, programmers often need to interfere with graphical designers' work and vice-versa, often resulting in a reduction of productivity. The purpose of Cocoon is to separate productive contexts to maximize the effectiveness of each team; style construction develops in parallel with logic design, improving productivity, quality and maintenability.

As per Web applications, the idea introduced by Cocoon is to use a pipeline to manage requests. A pipeline is a series of steps to process a particular kind of content. Usually, a Cocoon pipeline consists of a set of steps that

specify generation, transformation and serialization of SAX events composing generated content.

After being processed, requests move through pipeline stages. Each stage is responsible of a part of generation or transformation of contents. Cocoon allows to define all parts of a pipeline. SAX[30, 2] events are interposed between one phase and another; as an instance, the result of a pipeline can be a HTML page produced from a XML document.

A pipeline can be composed of four or more components always executed in the same order. For an example of pipeline see Fig. 1.

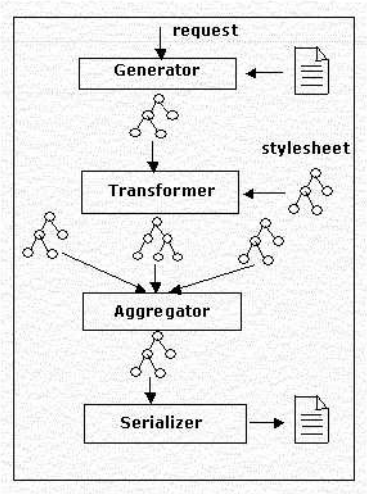

Fig. 1. A typical Apache Cocoon pipeline, from [18]

The sitemap is the heart of Cocoon. Here the developer configures Cocoon components and defines the client/server interactions in the pipeline. Cocoon matches each HTTP request to relative content in the sitemap, so that every part of the application (e.g. an XSLT file) is submitted to the appropriate component; each of them carries out a precise task and communicates with the precedent and/or the successive one by means of a stream of SAX events, activated when documents to be manipulated are submitted to the parser. SAX model consists of a set of classes and interfaces; it concerns two components placed in succession inside a pipeline; the first one sends a set of events, the following one pays attention waiting for these informations.

Transformations may be very demanding in terms of resources from servlet engine. Text parsing and transformations application require, in fact, a large quantity of processor resources. As to memory management, the situation has improved since Cocoon has adopted SAX in place of DOM[29], but this aspect is still problematic.

2.2 XQuery

XQuery is the language designed to query XML documents using XPath expressions. It's really a recent recommendation, become such through W3C on January 2007. It's not a fault to affirm that, from a semantic point of view, we are in front of a SQL for XML databases, as its aim is just this. XQuery syntax, however, is distanced from that one of its corresponding for relational databases: XQuery is, in fact, a procedural language made of functions (importable by means of namespaces), conditional and iterative instructions. The heart of language resides into **FLWOR** expressions, a set of five clauses (whose initials make the acronym) similar to that ones that form a SQL query:

- *For* assigns to a variable a list of elements, extracted from a XPath expression, involved in the XQuery query;
- *Let* operates a generic assignation (e.g. variable function value);
- *Where* establishes the condition to satisfy in the query;
- *Order* by establishes how results will be ordered;
- *Return* indicates the result of the query.

The argument of a clause is an expression in which function and XPath[22] expressions coexist. XQuery allows to embed code fragments inside HTML tags, on condition that they are delimited by braces. This feature permits to carry into effect, inside the same code and avoiding to recur to XSLT stylesheets, the separation between obtained data and their visual return.

2.3 XSP

XSP (eXtensible Server Page) is a language developed for Cocoon (by Cocoon developers) to create dynamic Web pages. It's still a technology under development, supported exclusively by this framework and composed of XML pages characterized by special tags. XSP programming is based essentially on three key points, through which separation between content and presentation is accomplished:

- use of tag libraries (logicsheets) imported by namespaces;
- use of a programming language (usually Java) inside appropriate markup elements;
- transformation of generated contents through XSLT stylesheets.

Each XSP page is processed by ServerPages generator, which represents in Cocoon the starting point of elaboration by means of pipelines. The ServerPagesGenerator transforms tags in a Java class which implements the Generator interface. XSP page is only compiled after first creation of the Generator; following executions will use the generator already available.

Each XSP pages starts with the $< xsp : page >$ tag; on its interior we declare the embedded programming language and the namespaces used to import tags from logicsheets. XSP supports programming language such

as Java, Javascript and Python. The rest of page comprises tags extracted from libraries and one or more $< xsp : logic >$ elements containing embedded code. XSP default library provides a further top-level element, called $< xsp : structure >$, in which declarations inherent to the used embedded language can be enclosed. Generally, it is used to declare the import of external modules as, for example, classes package. Being both logic and structure top-level elements, it's impossible to include one into the other.

Summarizing, an XSP page with only elements from default logicsheet introduces the following structure: a $< xsp : page >$ node, one or more $< xsp : structure >$ nodes, and one or more $< xsp : logic >$ nodes. The elements taken from this library don't allow a fluid XSP programming as they leave the development of dynamic content to embedded code, thus weighing down source code remarkably. Besides, it's advisable to divide code in syntactic markup blocks, each of them having its own function (session management, parameters management, etc.) and to commit what cannot be manipulated with these blocks to embedded logic or through creation of new specific logicsheets.

3 Our Project

The idea at the heart of our project [15] is to consider the Web as a huge database, each site representing an independent component which continuously generates updates. Thus, we face a multitude of information incessantly changing. It is also (more or less) homogeneously distributed on the whole network. Our goal is to retrieve RSS/Atom feeds published by some Web sites, store them in a Native XML Database (NXD) and publish them aggregated according to some filtering criteria, e.g. for thematic similarity. With respect to other Web-based feeds aggregators, we are able to submit XQuery queries to our repository, thus exploiting both the power of XQuery/XPath and the structure of RSS/Atom feeds. In order to publish feeds, we maintain a list of news sites which are frequently updated in order to retrieve fresh news. Our users can submit the URL of sites of her interest so to include them in our list. In order to enhance the performances of our portal, we decided to implement a caching mechanism, able to remember both the requested Web resources and the queries submitted by the users.

Indeed, each external URL access involves latency periods related to the nature of the connection. They grow linearly with the number of resources accessed.

A cache that memorizes the examined resource and the search parameters, has been used in order to eliminate this bottleneck. A deadline is assigned to each temporary version of the resources. It is defined as the parameter inside the pipeline, at the end of which the resource is considered stole. An additional Cocoon component (written in Java and inserted as a JAR) has been created in order to schedule cache updates. This "daemon" is like an

Action component inside the site-map. Moreover, the parameter concerning the duration of the cache is transferred.

3.1 Caching of Resources

Resources are served from an internal applicative pipeline which returns them through redirection, following a matching strategy studied for URLs that are corresponding to RSS and Atom files. This solution allows the storage and access to temporary copies of the requested resources, without causing modifications to the portal structure. Thus, together with the site-map, it defines an interface between our application and the Web.

A Java module has been implemented in order to schedule the access and then the storage of all resources in the cache through the creation of a connection and the request of an URL like http://www.feeding.it/allresources, which makes reference to a XQuery, forcing the update. During the first updating request a thread is created. It is kept in memory in order to satisfy the following updating requests and executed in parallel to both Java modules and searches. The response time is close to zero. If a search is executed during the updating operation, previously cached copies would be served.

3.2 Scalability Tests

A sequence of tests has been executed in order to study the speed of resources retrieval. The tests were made without using the cache, to better understand the updating times, with particular attention to differences of performance among searches inside and outside Italy. Each test has been made with 25, 50, 100, 200 resources and has been repeated ten times using the word "Iraq", first over Italian resources and then on non-Italian sites. Figures 2 and 3 illustrate the results of our test.

The max semi-dispersion here is represented by intervals of uncertainty enclosed within the upper and lower extremities which are respectively green and red. The rising of the resources coincides with the rising of intervals of semi-dispersion and a sub-linear growth of response times in the case of searches within Italy. We can note, moreover, how the response time for search done within Italy is extremely lower than that over the whole Web. This shows that latency times within the server determine performances.

3.3 Creating the portal

The portal has been called *Feeding* from the noun "feed" associated with the English suffix "ing" used to indicate action in progress, thus reflecting the nature of the project that handles data in continuous evolution. With the exception of thematic pages and search engine written in Xquery, the rest of the dynamic pages which compose the portal have been realized in

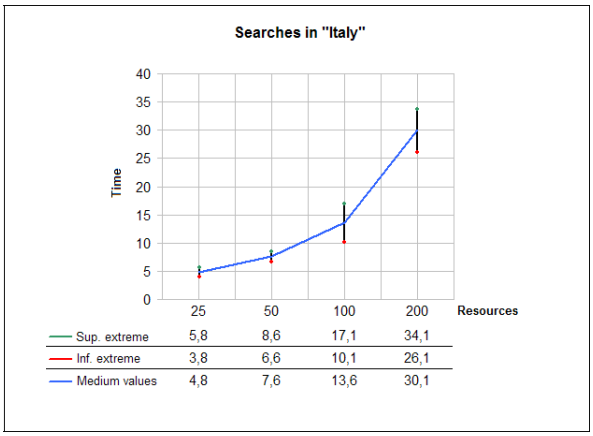

Fig. 2. Retrieval times vs no. of resources. Italian resources

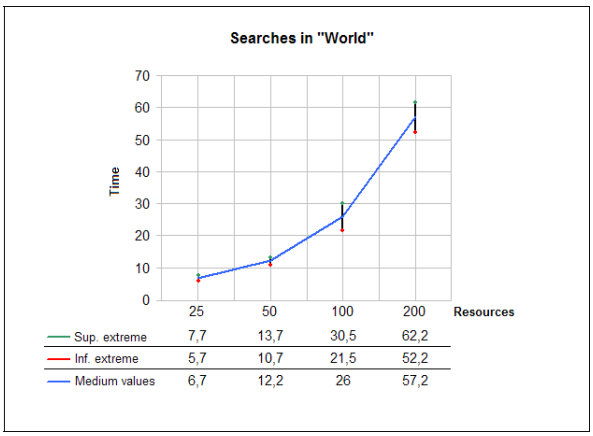

Fig. 3. Retrieval times of resources vs no. of resources. Non Italian resources

XSP and acquire the respective contents through SQL query on HSQL-DB, a RDBMS integrated in Cocoon. Queries are embedded in the XSP tags, then transformed through XSLT and XSL-FO style sheets. The native XML-DB, Exist, offers the Xquery support used to query RSS/Atom feeds. The graphic interface is written in CSS 2.0, XHTML-Transitional 1.0 and AJAX.

3.4 Creating the search engine

The search engine has been entirely written in Xquery. *Feeding* uses an XML file with the URLs of each feeds. Two attributes, which indicate the "language" and the "topic", have been assigned to each URL, respectively. When search is done, the selected key-words and the radio-button index are transmitted by URL-rewriting to the page of interest. The news source acts as a

filter in order to select the XML-path used in the query. It is in fact the result of a Xquery function that uses the above parameters as arguments. The main function of the engine uses the search keys obtained and returns all the occurrences within the elements *item/title* and *item/description* of the feed. This procedure is iterated for all RSS/Atom resources of interest. The news of each feed are then listed ordered by publication date.

Feeding allows the use of advanced functions in order to obtain a highly selective search. Selection criteria may be specified in one of these forms:

- Basic search: *Università Messina*. Looks for the first OR the second word occurrence in the feed.
- Pattern search: *"Università di Messina"*. Looks for the exact pattern occurrence in the feed.
- Exclusion search: *Università -Messina*. Looks for Università without Messina matching.
- Inclusive search: *Università +Messina*. Looks for both Università AND Messina, meaning that the feed has to have both words at once.
- Search with date: *2007-04-07*. Looks for all news published in the specified data. The data has to be written in the form YYYY-MM-DD, MM and DD can be optionally excluded.

3.5 Complex searches

One or more of the listed search forms can be used at the same time, thus allowing the users to make complex searches as: *Università -Messina 2007-03-07*. It looks for all news published in the specified data with the matching word "Universitá" and without "Messina". Even better, a user can search for terms appearing in the *title* field, other terms appearing in the *description* field while limiting the feeds only to those published within a time period, say a couple of days.

3.6 The thematic pages

The content of each thematic page is generated through a query which acquires its parameters through URL rewriting and uses it to select the resources from which the news will be extracted. We notice that each URL inside the XML file is equipped with an attribute that specifies the topic of the feed. The content of the page is generated through a query which associates the acquired parameter during the request according to the value of the attribute mentioned above. The result of the query includes the latest news published for each feed.

4 Related work

Apache Cocoon is a successful framework and by now it has been deployed at several sites[1]; some of which exploit its main feature, that is the separation of content, logic and presentation. In the few last years also some scientific projects adopted Apache Cocoon as a framework for their applications, even if their field of interest differs from ours. See [1, 9] for sample applications.

As to the main goal of our project, that is management of repositories of RSS/Atom feeds and the subsequent extraction of relevant information, we found a correspondence in the works on information extraction tools. These, whose aim is to convert semi-structured or structured Web content into a structured, i.e. XML, format, have been thoroughly surveyed from a number of authors. See for example [36, 37, 38] and references therein.

5 Conclusions and future work

We presented a new platform for retrieval and querying of RSS/Atom feeds by means of a powerful XQuery engine, which fully exploits the structure of XML documents. Selected RSS/Atom news sites are frequently queried and newly produced feeds are retrieved and stored in a local XML database for future queries. Although our project is still in a early development stage, its first results seem promising and the emphasis on Xquery queries are unique among various feeds portal on the Web. We planned, as our next achievings, to better manage feeds polling to minimize the number of unnecessary feed retrievals and to publish our platform on the Web.

References

1. Eidenberger H (2004) Modelling of Visual Feature Derivation in the Vizir Framework. Proceedings European Signal Processing Conference, Vienna
2. Faragas L (2004) The Joy of SAX. First International Workshop on XQuery Implementation, Experience and Perspectives, Paris, France
3. Ford N (2003) Art of Java Web Development: Struts, Tapestry, Commons, Velocity, JUnit, Axis, Cocoon, InternetBeans, WebWork. Manning Publications
4. Hammersley B (2005) Developing Feeds with RSS and Atom. O'Reilly Media, Inc.; 1 edition
5. Jafari A (2003) Designing Portals: Opportunities and Challenges. Information Science Publishing
6. Johnson D (2006) RSS and Atom in Action: Web 2.0 Building Blocks. Manning Publications
7. Kraus A, Koch N (2002) Generation of Web Application from UML Models using an XML Publishing Framework. 6th World Conference on Integrated Design and Process Technology, Pasadena, CA

[1] See http://cocoon.apache.org/link/ for an updated list

8. Leung T W (2003) Professional XML Development with Apache Tools: Xerces, Xalan, FOP, Cocoon, Axis, Xindice. Wrox
9. Madeyski L, Stochmialek M (2004) Architecture of Modern Web Application. Software Engineering after the year
10. Melton J, Buxton S (2006) Querying XML: XQuery, XPath, and SQL/XML in context. Morgan Kaufmann
11. Moczar L, Aston J (2002) Cocoon Developer's Handbook. Sams; 1st edition
12. Robie J (2003) SQL/XML, XQuery, and Native XML Programming Languages. XML Conference and Exposition, Pennsylvania Convention Center, Philadelphia, PA
13. Sangmi L, Sunghoon K, Fox G (2003) Adapting Content for Mobile Devices in Heterogeneous Collaboration Environments. ICWN Cocoon
14. Ziegeler C, Langham M (2002) Cocoon: Building XML Applications. Sams; Pap/Cdr edition
15. La Rosa M, Pimpo T (2007) Ricerca di feeds RSS/Atom su database dinamici distribuiti: un portale con il framework Cocoon. Graduation project. University of Messina
16. Wittenbrink H (2005) Rss And Atom: Understanding And Implementing Content Feeds And Syndication. Packt Publishing
17. Garrett J J (2005) Ajax: A New Approach to Web Applications. http://www.adaptivepath.com/publications/essays/archives/000385.php
18. Apache Cocoon Project http://cocoon.apache.org
19. W3C CSS 2.1 Specs http://www.w3.org/Style/CSS/
20. Sun Java Enterprise Edition http://java.sun.com/javaee/
21. W3C XHTML 1.0 Specs http://www.w3.org/TR/xhtml1/
22. W3C XPath Specs http://www.w3.org/TR/xpath
23. W3C XQuery 1.1 Specs http://www.w3.org/XML/Query/
24. W3C XSL http://www.w3.org/Style/XSL/
25. W3C Xslt http://www.w3.org/TR/xslt
26. Apache Cocoon Project - XSP http://cocoon.apache.org/2.1/userdocs/xsp.html
27. Apache Avalon model http://cocoon.apache.org/2.1/developing/avalon.html
28. Apache Excalibur Project http://excalibur.apache.org/
29. W3C DOM http://www.w3.org/DOM/
30. SAX Project http://www.saxproject.org/
31. Hursch L, Videira Lopes C (1995) Separation of Concerns. TR NU-CCS-95-03, College of Computer Science, Northeastern University, Boston, MA
32. Kener C, Kirda E (2000) Layout, Content and Logic Separation in Web Engineering. 9th International WWW Conference, 3rd Web Engineering Workshop, Amsterdam
33. Burner A (2002) Comparison of Web Technologies and Web Engineering Methodologies. BurnerNet.com
34. Reina A M, Torres J, Toro M (2003) Aspect-Oriented Web Development vs. Non Aspect-Oriented Web Development. Workshop of analysis of Aspect-Oriented Software, Darmstadt, Germany
35. Aksit M (1996) Composition and Separation of Concerns in the Object-Oriented Model. ACM Computing Surveys
36. Laender A.H.F. , Ribeiro-Neto B.A., da Silva A.S., Teixeira J.S. (2002) A Brief Survey of Web Data Extraction Tools SIGMOD Records 31(2) 2002
37. Flesca S., Manco G., Masciari E., Rende E. and Tagarelli A. (2004) Web wrapper induction: a brief survey. AI Communications 17 (2004) 57 - 61

38. Chia-Hui Chang, Kayed M., Girgis M.R., Shaalan K. (2006) A Survey of Web Information Extraction Systems IEEE Transactions on Knowledge and Data Engineering, TKDE-0475-1104.R3

Virtual Communities as Narrative Processes

Marco Benini and Federico Gobbo

Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
via Mazzini 5, IT-21100, Varese, Italy
{marco.benini, federico.gobbo}@uninsubria.it

Abstract. By facing the problem to describe the history of a virtual community as the sequence of events generated by its participants, a different perception of the meaning of communitywares emerges. This paper describes a proposal for a virtual community system based on the narrative process that supports the social evolution of the community.

Key words: Communityware design, Semantic web technology, Ontologies

1 Introduction

Discussion and collaboration servers, i.e. software tools that promote and mediate dialogue and partnership among different users, are not a novelty anymore: they gradually grew following the network spread and evolution [1]. Nevertheless, there is still no tool supporting the evolution of the rules of the social network underpinning cooperation. In fact, the social rules are mostly defined by designers, and hard-coded into the collaborative system, without explicit semantic information. In this paper a formal model to represent the most known collaboration models is given, in terms of semantic web ontologies. These ontologies will be managed in natural language, so that users can define by themselves new, original forms of cooperation and dialogue.

In principle, there are three basic collaboration models over a network: e-mail exchange (including mailing lists), shared repositories, and interactive content update technologies [7]. Virtual communities, encouraging participation and active learning among remote users, naturally prefer the third model since their members aim to establish social relations, and this goal is easier to achieve if users are allowed to update content interactively.

As a matter of fact, virtual communities evolved around complex communitywares which combine the feature of the three basic models. In fact, their main service [1] was to provide discussion lists, often called *conferences*, where the participants were allowed to discuss over common topics (the e-mail

exchange model), while additional features were usually provided, as shared repositories (i.e. the second model), or personal web pages, email address, etc. The aim behind these systems was to offer an all-inclusive environment [11], in order to give a complete support to each participant's need, so that the community members were invited to use the Internet almost exclusively through the community support.

Henceforth, as communities evolved, the software platforms grew in complexity, due to the subsequent addition of unplanned features. In fact, it is very difficult, if not impossible, to foresee every participant's need or desire in advance, i.e. before the virtual community establishes itself, as people expectations are usually very different: our claim is that these wishes cannot be foreseen since they arise **after** the community uses the software for enough time to evolve itself, while the design of the software takes place **before** the community starts to operate.

2 Virtual communities and "new texts"

Since the end of the 20^{th} century, the increase of network size and speed and the standardisation of the web [2] also lead to a deep transformation of virtual communities. Community services became differentiated according to their needs. In our opinion, the deep reasons behind this fragmentation lie in the increase of users' awareness: the Internet services are now mostly well-known and, thus, users don't need an active guidance in their usage anymore. Indeed, it is not surprising the raise in popularity of a new kind of community-oriented services, broadly called *new texts*, like for instance *wikies*, which allow the collaborative development of knowledge, or *blogs*, which act as discussion vehicles [5]. However, despite their maturity as technological objects, the design of communitywares and new text services is similar and still quite traditional: they are usually developed as specialised web-based applications [11].

Their design and development is focused on the web technology and its clever application to the problem domain; the simple idea that the purpose of the software is **just** to support a living community is left in the background. In the approach proposed here, the reversal is true: a communityware should support a virtual community from its start permitting its evolution with the social rules that participants arbitrarily decide to adopt, according to the community life. Moreover, the social rules belong to the community, which can modify them over time to reflect new needs and wishes.

2.1 Blogs and wikies as narratives

Since our aim is to propose a reversal approach, in the following we will describe ideal "new text" communitywares and how their core works. A designer may either directly implement this approach, or, preferably, may include techniques and ideas in a richer system, where the features avoided for clearness and conciseness are present and fully supported.

We start by designing and thus constructing a language allowing the writing of the community history. Hence we call our approach *narrative*, since virtual communities are considered as narrative processes. This narration is described by means of a language, which has enough expressive power to depict also the community state, that is, the information owned by the community. In this perspective, the language itself is part of the state; since the state varies over time, and the language is part of it, the language may evolve as well. As far as the features used by the community are defined by the language, any addition to the language corresponds to an evolution of the community in terms of represented features, thus overcoming the discussed ageing problem.

In order to concretely exemplify our ideas we define the words "User", "Message" and "Conference". Their intended meaning[1] is as follows: the users are the actors of the community, i.e. they can perform actions like sending messages, and, in turn, messages are organised to form conferences. The community state is the sum of the conferences and the language defined insofar. The community history tales the changes in the community state.

Communitywares based on the e-mail exchange model [7] – e.g. BBS, mailing lists, web forums, web groups – organise content on the paradigm "write once, read many". In fact, in this paradigm, conferences are *threads*, owned by no user in particular. A message, the *root*, starts a thread on a specific question or topic, which sequentially people answer or comment. If a message is off-topic, a new thread begins. Threads are often very long, and the result is a complex tree of messages, where conference boundaries are not always clear as messages belong to more than one conferences, and redundancy in the messages content is tolerated [7].

Blogs are a significant variant of this paradigm, which we call the *annotation model*. In fact, unlike what happens with mailing lists, blogs have a clearly defined author – maybe collective, but still one – who owns the conferences and has the right to manage their messages. Conferences are shaped as threads, but the root (called *post* in the blog jargon) is more important than the threaded answers, which can be considered as mere comments. Unlike mailing lists, threads are usually short, and not rarely they are made by a unique message, the post. Comments are not the only way to answer to one's post: blogs are by no means living as monads, on the contrary, *annotations* – i.e. messages belonging to a blog but pertaining to another blog post – are allowed and encouraged. When annotation happens, blogs are put into relation and form a *blogosphere* – another form of community [10]. Fig. 1 shows a prototypical example: John has raised an issue (post B) for further considerations on Tuesday in his blog, and Pietro reacts commenting it in John's blog space. On the contrary, Mario, after reading B, decides to write the longer answer D as an annotation of B, perhaps via a citation. Thus, on Wednesday John's and Mario's blogs are intertwined. In this picture, Jack is allowed to comment but he decides to read without reacting. In blogs, the

[1] These notions are standard and described at length, e.g., in [1].

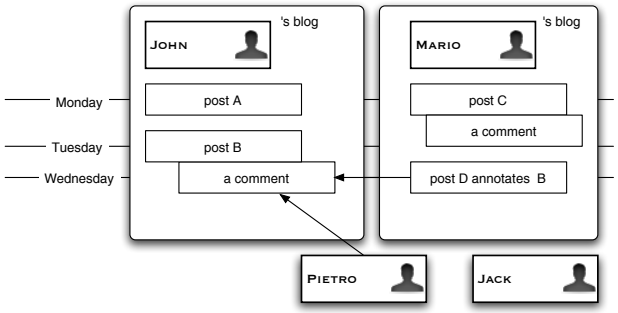

Fig. 1. A minimal blogosphere

content is organised on the paradigm "write yours, read and comment the others". In the terms given above, a blog is a set of conferences owned by a user with a defined identity. Wikies are quite different from blogs, as in the

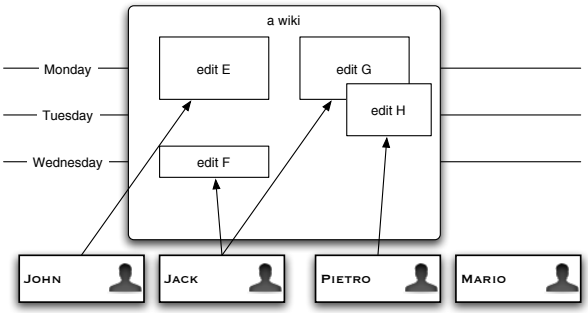

Fig. 2. A pure wiki

shown example (Fig. 2). On Monday John and Jack start two different wiki pages (edit E, G) on some related topics – in our terms, they had created two conferences. Note that in a pure wiki no message is authored, i.e. all messages are anonymous. Pietro reads Jack's message on Tuesday, and he writes a message (edit H) that updates the message body. Jack, who evidently likes the wiki way more than blogs, reads the changes and decides to add some information to the conference started by John, appending some content (edit F). The conference history becomes a sequence of patches of differences between subsequent messages.

3 From natural language to ontologies

When describing a blog or a wiki, it is natural and easier to depict an example of the intended model, as done in the previous section.

One can try to move toward a formal description by *narrating* the example: "John is an user. John's blog is a set of conferences, owned by John. A comment is a message. Only users may post messages". This informal description identifies some social rules, some entities and some roles: John and his blog are entities, they gain a social role by their attributes, like being a user or a blog. There are other, unidentified entities, like messages, and they can be related to known objects by some actions, like "post", whose usage is restricted to users when involving comments.

In this perspective, narrating the history of the community means to record the sequence of actions performed in the community world. Every action is composed by a series of *events*, each one described by a simple sentence. When able to interpret the meaning of events as actions to apply on the actual community information, the community will be enlivened by a suitable engine that receives and performs the action on the community state. If narratives, e.g. the example before, are formalised into events, it becomes feasible to develop an engine for a narrative communityware.

Our investigation on the informal description starts by analysing the sentences: sentences are structured groups of elements, where each element plays a role or a defined function – hierarchically defined. According to Tesnière's structural grammars [12], a sentence is a set of connections, where its type is defined by the verb: the term *valence* refers to the number of arguments, or *actants*, a verb can take. For example, the sentence "John owns John's Blog" contains "owns" which is a divalent verb, i.e. it has a first actant ("John", the subject) and a second actant. On the contrary, the verb "to be" is monovalent, as it has only one actant (the subject), and denotes an attribute of it. In order to avoid unnecessary complexity in natural language parsing, only present tense is used, i.e. sentences are all statements. Besides verbs, there are nouns: generally speaking they denote either concrete entities, like "John", or concepts, like "user". Some verbs and some nouns are predefined, i.e., their meaning is common knowledge. These elements are "to be" and "may" in our example. On the contrary, most nouns and verbs have a specific meaning which depends on the particular community we are constructing: "user", "conference", "to own" and "to post" are of this kind, since their interpretation varies if the community is a wiki, a blog or something else. Therefore, a formal description must define these notions and a communityware engine has to provide a model, enabling their subsequent use on the community state.

The description we propose is based on a pair of knowledge bases, represented as OWL ontologies: the *history* and the *state* of the community. The history contains the recording of the sequence of events occurring during the community life. The state contains the language definitions and the information owned by the community as it holds in a particular instant; while the

history is constantly growing, the state gets updated by every event. In this respect, using an ontology to represent the state allows both to dress the language with a logical meaning and to ensure the formal consistency of the depicted community world moment by moment.

3.1 Sketches from a Narrative Community

The narrative approach can be formalised in an operative model of the previous examples: we start by defining a simple language that allows the narration of the community events; the events will be the actions each participant performs in the community. The syntax is based on a set of nouns and verbs that allows the constructions of simple sentences: for convenience, we use the OWL syntax [4, 8][2] that simplifies the understanding of the system's behaviour.

In the beginning, the history of the community as well as its state are empty, and the language is pure OWL plus the `vcs` (*virtual community structure*) namespace, whose content is explained later. The first step is to define the notions of "User", "Message" and "Conference". A user, the community starter, narrates the following events to the system:

```
<owl:Class rdf:ID="Noun" />
<owl:Class rdf:ID="User">
  <rdfs:subClassOf rdf:resource="#Noun" />
</owl:Class>
<owl:Class rdf:ID="Message" />
  <rdfs:subClassOf rdf:resource="#Noun" />
</owl:Class>
<owl:Class rdf:ID="Conference" />
  <rdfs:subClassOf rdf:resource="#Noun" />
</owl:Class>
```

A "Noun" is rendered as an OWL class; "User", "Message" and "Conference" are nouns. Analogously, he can describe the basic verbs to interact with the concepts just defined:

```
<owl:Class rdf:ID="Verb">
  <rdfs:subClassOf rdf:resource="&owl:ObjectProperty" />
</owl:Class>
<Verb rdf:ID="read">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Message" />
      <owl:Class rdf:about="#Conference" />
    </owl:unionOf>
  </rdfs:range>
  <vcs:action> ... </vcs:action>
```

[2] We assume the standard conventions for namespaces in OWL fragments, see [8].

```
</Verb>
<Verb rdf:ID="own">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="#Conference" />
  <vcs:action> ... </vcs:action>
</Verb>
<Verb rdf:ID="post">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="#Message" />
  <vcs:action> ... </vcs:action>
</Verb>
```

Therefore, a verb like "read" is both a linguistic element in the "Verb" class, and an OWL-property whose domain (the subject of the verb) is a "User" and whose range (the object of the verb) is either a "Message" or a "Conference". Consequently, "read" has a triple meaning: as a linguistic element, it is a bivalent verb; as an action, it denotes the transformation on the state as calculated by its `<vcs:action>` tag; as an OWL element, it is a property relating class elements. In particular, the `<vcs:action>` declares the effect of the verb on the state of the community by means of a program written in XML/XQuery [3], linked via the `vcs` namespace – details are omitted here for clarity. The state of the community is an OWL ontology containing the sentences in the defined language that describe the information of the community. The action is a piece of programming code that defines the change on the state when a related event happens: for example, when the event "John posts the message X" occurs, the message X is added to the community state by the program contained in the `<vcs:action>` of the declaration of the "post" verb (see next subsection for a precise description).

To describe the structure of a message, we enrich our language with attributes, represented as OWL datatype properties:

```
<owl:DatatypeProperty rdf:ID="title">
  <rdfs:domain rdf:resource="#Message" />
  <rdfs:range rdf:resource="&xsd:string" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="content">
  <rdfs:domain rdf:resource="#Message" />
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="inConference">
  <rdfs:domain rdf:resource="#Message" />
  <rdfs:range rdf:resource="#Conference" />
</owl:ObjectProperty>
```

To show how the previous declarations can be used, we populate the state with some facts, from the examples in Sec. 2.1:

```
<User rdf:ID="John" />
<Conference rdf:ID="JohnBlog" />
```

```
<Message rdf:ID="msg1">
  <title> Post A </title>
  <content rdf:resource="http://www.dicom.uninsubria.it" />
  <inConference rdf:resource="#JohnBlog"/>
</Message>
<User rdf:about="#John">
  <own rdf:resource="#msg1" />
</User>
```

The narrative approach, as described till now, allows both to write the history of the community, and to operate the core actions on the community state. Moreover, the language used to tale the events is defined as part of the narration, like in mathematical textbooks, where the concepts are first defined, and then used to derive results and to define new notions.

In the emerging model, nothing prevents the reflective usage of already defined concepts. For example, we can define a conference whose elements are the defined users. The event we submit to the system is the following:

```
<Conference rdf:ID="Users" />
<owl:Class rdf:about="#User">
  <rdfs:subClassOf rdf:resource="#Message" />
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#inConference" />
      <owl:allValuesFrom rdf:resource="#Users" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

It means that a user is a special kind of "Message", which lies in the "Users" conference. The result is that user management does not require new verbs or special actions: the ability to post in the "Users" conference allows the creation, cancellation and modification of the set of users by means of the very same actions used to manage any other message. An important point to notice is that we **evolved** toward this kind of management: in fact, we incrementally derived the idea of managing the users adding a new conference to an existing community where, initially, "User" was a standalone concept.

The reflective use of concepts is nothing but an example of evolution: in fact, since the language may be modified at any time, potentially every event involving a change in the language can be regarded as a step toward the evolution of the community.

4 Behind the Curtain

The ideal communityware engine implicitly used in the preceding section is very simple: it takes its input, an *event*, from the web, processes it and calculates the output, usually an XML document. The event is an OWL fragment,

that must be understandable in the current state, that is, the state ontology plus the event must form a valid XML document as defined in [13], satisfying the OWL syntax augmented with the `vcs` namespace.

Moreover, the event must be semantically sound with the state, that is, the state ontology plus the event must form an OWL-consistent document as defined in [9], thus generating a logically sound theory.

If the event is both valid and sound, it denotes the actions that must be performed on the ontology state: each action is defined by means of a function written in XML/XQuery [3], represented inside the definition of the simple event's verb via the `<vcs:action>` tag; the default action (when the tag is missing) is to append the event to the ontology state. Therefore, the denoted action is tentatively performed and the resulting state is checked to be valid and sound. As a consequence, the output is calculated as the updated state ontology: in a real system, this would be inappropriate and a suitable presentation of part of the state should be extracted and shown to the user. Finally, the event is recorded in the community history.

Although heavily based on the semantic web technologies, the described engine operates as a simplified web-based application. But, differently from the traditional communitywares, wikies and blogs, it does not provide hard-coded notions, actions and rules. As previously illustrated, even the basic notions, like *user* or *message*, are defined "in the language" and, thus, they become part of the state hence, as any other element of the state, they may be modified, created or cancelled with the only limitation that the resulting state preserves validity and soundness, i.e. it has to be a well-formed OWL ontology with no internal contradictions.

As a matter of fact, the abstractness and the generality of the illustrated engine provide the community with the instruments to sustain its own evolution, since literally everything can be discussed and, eventually, modified. It is evident that, in practice, a more significant starting point is needed, that is, the initial language should be non-empty and should represent a well recognised language to describe a community model. In this respect, the shown sketch is too limited, but the discussion in Sec. 2.1 provide the highlights to develop the notions and, thus, the language constructors needed to represent the corresponding community models, starting from blogs and wikies.

In fact, the narration of an example of community life requires a language that can be usefully represented in the form of an OWL ontology; this ontology becomes the foundational event of the community, enabling its usage by means of the illustrated engine. Therefore, the narrative description of communities becomes the enabling metaphor that allows their representation in a semantic web system. Because of the expressive power of semantic web conceptual instruments, it is possible to enliven the narrative representations of communities in order to support them and, eventually, their evolutions.

5 Concluding remarks

This paper has shown the idea that considering virtual communities as the result of a narrative process, leads to a new possible design approach of the communitywares to support them. This approach wants to suggest that the semantic web technology is mature enough to permit a significant encoding of virtual communities in its main representation language, namely OWL. In this respect, ontologies become the instrument both to represent and to operate communities, with a degree of freedom and flexibility unachievable in traditional and modern communitywares.

Being a proposal paper, a great deal of future work is expected: in the first place, the implementation of the engine and the consequent collection of experimental data. Also, it is important to study to what extent reflection (see the end of Sec. 3.1) can be used to simplify the management of complex communities. Finally, the proposed approach allows to simulate communities with specific social rules: for example, the study of the application of the rules formalising Creative Commons licenses [6]. Although we have begun to explore some of these themes [5], most is still to be done.

References

1. M. Benini, F. De Cindio, and L. Sonnante. Virtuose, a VIRTual CommUnity Open Source Engine for integrating civic networks and digital cities. In P. van den Besselaar and S. Koizumi, eds, *Digital Cities III — Information Technologies for Social Capital: Cross-Cultural Perspectives*, volume 3081 of *LNCS*, pp. 217–232. Springer Verlag, 2003.
2. T. Berners-Lee. *Weaving the Web*. Harper, 2002.
3. D. Chamberlin et al. *XQuery from the Experts*. Addison Wesley, 2003.
4. M. Dean and G. Schreiber. OWL web ontology language reference. W3C, Feb. 2004.
5. F. Gobbo, M. Chinosi, and M. Pepe. Novelle, a collaborative open source writing software. In J. Karlgren, ed., *NEW TEXT: Wikis and blogs and other dynamic text sources*, Trento, Italy, 2006. Association for Computational Linguistics.
6. L. Lessig. *Free Culture*. Penguin, 2004.
7. B. Leuf and W. Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison Wesley, 2002.
8. D.L. McGuinness and F. van Harmelen. OWL web ontology language overview. W3C, Feb. 2004.
9. P.F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL web ontology language semantics and abstract syntax. W3C, Feb. 2004.
10. Andrew Rosenbloom. The blogosphere. *Comm. of the ACM*, 47(12), Dec. 2004.
11. D. Schuler. New communities and new community networks. In M. Gurstein, ed., *Community Informatics: Enabling Communities with Information and Communications Technologies*, Hershey, USA, 2000. Idea Publishing Group.
12. L. Tesnière. *Éléments de syntaxe structurale*. Klincksieck, Paris, 1959.
13. F. Yergeau et al. Extensible markup language (XML) 1.1. W3C, Feb. 2004.

Bridging Different Generation of Web via Exploiting Semantic Social Web Blog Portal

Yuh-Jong Hu and Cheng-Yuan Yu

Emerging Network Technology (ENT) Lab. Dept. of Computer Science
National Chengchi University, Taipei, Taiwan, 11605, (hu, g9302)@cs.nccu.edu.tw

Abstract. The goal of this research is to analyze one of the Web 2.0 platforms, e.g., weblog (or blog) and to justify whether it is possible to bridge Web 2.0 ↔ Web 3.0 (or the semantic web) via exploiting semantic social web blog portal. Compared with semantic annotation system using web mining techniques to extract keywords from the WWW, our semantic social web annotation system is based on ontologies derived from folksonomy tagging system to truly reflect the intentions of people on the classification of resources. The blogsphere will be our first experimental example to validate the ontology+folksonomy mashup model. We hope this idea can be applied to the other Web 2.0 platforms, such as wiki, web services. We have built a semantic social web blog portal from the Taiwan's biggest blog service provider (BSP). From this semantic social web blog portal, users are allowed to execute a variety of online semantic social web queries that can not be achieved from other Web 2.0 blog search engines, such as Blogpulse or Technorati. The incentives of having semantic social web annotation for blogsphere were justified and this might shed some light on bridging Web 2.0 ↔ Web 3.0.

1 Introduction

The principles on how to identify one application as "Web 1.0" and another as "Web 2.0" were previously clarified by Tim O'Reilly [21]. The Web x.0 indicates how the x.0 Web generation platform copes with their contents writer and reader's experiences. The bridging of Web generation is defined as the contents created in previous generation of Web can be extracted or accessed in next generation (or vice versa). The bridging of Web 1.0 ↔ Web 2.0 is an ongoing process while the bridging of Web 2.0 ↔ Web 3.0 is not well understood yet. To bridge different Web generation does not necessarily mean the old generation Web will be completely phased out. On the contrary, different Web generation still might be happily live together. Ever since the New York Times reporter John Markoff coined the semantic web as Web 3.0 [16], we were curious whether there existed a feasible bridging mechanism for Web 2.0 ↔ Web 3.0 via integrating respective folksonomy and ontology technologies.

The folksonomy of tagging system for blogs is an example to enable social web services in the Web 2.0. On the other hand, ontologies with their machine understandable metadata aim at achieving Web 3.0 vision. If we can (semi-)automatically mash up the ontology data and query model with the folksonomy tagging system services, then we are in a very good shape toward this paradigm shift. Eventually, this might realize Tim Berners-Lee's semantic web vision for an extension of the current web (Web 1.0 or 2.0) in which information is given well-defined meaning and better enabling computers and people to work in cooperation [2].

In Web 2.0, people interact with each other and address their opinions voluntarily. The challenge of this social web services depends on whether we can collect these huge amount of unstructured public opinions and discover the patterns among them. For the past few years, research issues for the development of annotation system on bridging Web 1.0 $\leftrightarrow$ Web 3.0 were intensively investigated. People were trying to figure it out whether it is possibly to bridge existing Web 1.0 with the future semantic web (or Web 3.0) [7][8][14]. Unfortunately, the progress of this study seems to be very slow because it is a grand challenge to have (semi)-automatic semantic annotation system to create ontology-based semantic annotations from huge amount of unstructured WWW contents.

The social web annotation of bridging Web 2.0 $\leftrightarrow$ Web 3.0 seems to provide another window to deal with this problem. In social web annotation system, people use free tags (or vocabularies) to address their opinions or preferences on the Internet resources, such as bookmarks, videos, blogs, and web pages, without relying on controlled vocabularies. This resolves a hard design problem for the construction of agreeable monolithic heavyweight ontologies. Because it is more explicit and direct on the categorization of resources via free tags from folksonomy than keywords mining from the Web's contents [4]. Tagging systems are still not well studied and have the research potential for further improvement [17]. Are there any other incentives to use free tags social web annotation rather than to use conventional keywords-based annotation? This will be the issue we are interested to investigate further.

Ontologies are top-down approach with hierarchical classification of information sharing and manipulation mechanism while folksonomy is a bottom-up approach using flat indexing to organize and search information through user feedback. When we regard users' free tags as social web annotations, we still might need ontologies to classify these free tags into different taxonomy. Furthermore, ontologies can provide well-defined structure schema to bind entity semantic association together and that was impossible to be realized by the tagging system alone. These entity relationship semantics might exist among tagger, tags, and resources declared implicitly by entity themselves. We propose a blog ontology and a topic ontology to harbor all of these free tags to describe the semantics of entity relationships in the blogspace. We allow users to explicitly enable semantic social web query for tags with their entity semantic relationships to get what they are really interested.

In order to exploit the incentives of bridging different Web generation, we have built a semantic social web blog portal from the biggest blog service provider (BSP) *WRETCH* in Taiwan [1]. We have implemented blog crawlers to collect all of necessary context and content information from this BSP. Three kinds of information sources were collected for this study: semi-structured HTML blog pages, structured XML-based RSS, and users' annotation free tags. The content and context information from these sources were extracted, analyzed, and stored to satisfy user's later semantic query services. Furthermore, we also analyzed the blog information diffusion flow using social network analysis (SNA) to examine the possible patterns in the *WRTECH* BSP [24]. Therefore users are allowed to enable semantic social web query services using a variety of SNA measures in our semantic social web blog portal.

2 Related Work

Several important elements are required to exploit the bridging problem of Web 2.0 $\leftrightarrow$ Web 3.0 to have semantic social web search services. They are annotation, ontology, blog, folksonomy, and SNA. Unfortunately, most of the related studies shown as the followings did not have these comprehensive considerations so they can not have the service capabilities as ours:

- Semantic annotation for ontology+web: The semantic annotation (or bridging) of Web 1.0 $\leftrightarrow$ Web 3.0 were extensively investigated before to support the indexing and retrieval of well-defined semantic information for agents [14][19][22]. The goals of these studies were too ambitious to have any significant progress.
- Semantic tag for ontology+folksonomy: Gruber proposed the mashup of ontology and folksonomy to enable social web ecology on the Internet [10]. The tagOntology was a very primitive study for identifying and formalizing a conceptualization of the activity of tagging.
- Semantic blog for ontology+blog: Semantic blog systems were built to leverage the power of ontology data model so that people can extract all of the implicit semantics from blogs [3][5][13]. But they did not really work for lacking enough amount of real dataset to experiment the system's feasibility.
- Tagging blog for tags+blog: Brooks et al. analyzed the top 350 tags from the Technorati blog search engine and they demonstrated that tags are useful for grouping articles into broad categories but less effective in indicating the particular content of an article [4]. This study did not aim at solving the bridging problem of Web 2.0 $\leftrightarrow$ Web 3.0 either.
- Semantic Web (or Web) as social network: In a semantic social network, a number of electronic information sources including web pages, emails,

[1] http://www.wretch.cc/

FOAF profiles, are extracted and analyzed to acquire their semantic relationships [9][19]. The purposes of these studies were to apply SNA techniques to analyze the ontology-based context information for the semantic web research community.

- Blog as social network: Gruhl et al. studied the dynamics of information propagation through blogspace [11]. Furthermore, the blogspace can be shown as community using SNA model to express its entity social relationships through links, comments, and trackbacks, etc [1][6][15]. But they only addressed pure blog ecosystems.

3 Research Goal

The goal of this research is to construct a semantic social web portal and to exploit the incentives of bridging Web 2.0 $\leftrightarrow$ Web 3.0. The incentives will be justified when we can search information through this semantic social web portal compared with other systems that only provide simple tags (or keywords) search on Web 2.0 or ontology query on Web 3.0. Unless we can extend tags to have corresponding semantic context, the expressive power of tags is limited. In this study, we found that coherent taxonomies of blog articles can emerge from users tagging so relevant customized ontologies can be constructed.

3.1 Social Network Analysis

Social network analysis (SNA) is the quantity study of the relationships between individuals or organization. By quantifying social network structures, we can determine where are the most important nodes in the network [24]. The implications of SNA usage are quite different when we apply SNA to different generation of Web.

- SNA for Web 1.0: The information on Web 1.0 is rather static so people only apply SNA on paper citation network or person relationship network to discover their stable relationships [18].
- SNA for Web 2.0: The nature of information flow on Web 2.0 is dynamic and user oriented. All of the tags, resources, and tagger's profiles on Web 2.0 are dynamically created so the challenge to apply SNA for this platform is how can we timely extract the relationships between taggers with annotated tags and their respective resources to enable effective information search [12].
- SNA for Web 3.0: We are aiming at bridging of Web 2.0 $\leftrightarrow$ Web 3.0. The issues we consider including Web 2.0, Web 3.0, and SNA, are different from pure semantic social network approach shown in [19].

3.2 Blogs as Social Network

Applying SNA model to the blogsphere has revealed interesting findings about how individuals share information and interact socially online. Social relationships can be expressed online as different forms of blogs ties: blogroll links, citation links, and comment links [1]. We observed the *WRETCH* blog communities in terms of important SNA measures, such as indegree/outdegree, closeness/betweenness, and k-cores, to interpret their social implications. The basic idea is that blog article written by important blogger also becomes important itself so we can reinforce the semantic search service capabilities for users to satisfy their interested from this perspective idea.

- Indegree and Outdegree: The higher indegree measure indicates the higher spread of blogger (or article) influence in the blogsphere. The indegree measures of the top 300 bloggers in the *WRETCH* BSP were shown as power law distribution. Contrarily, outdegree measure did not indicate any importance of a blogger in the community and its pattern did not appear as power law distribution either.
- Closeness and Betweenness: The higher closeness (or betweenness) of a blogger means it is in the social network center (or pivoting bridge) so the spread of influences of this blogger is significant in the community. We found that closeness (or betweenness) is similar to indegree but it incurs high computation overhead so we avoid computing this measure in our online information access.
- K-Cores: A k-core is a maximal subgroup in which each blogger (or article) has at least degree k within the subgroup. Thus k-cores measure is effectively to demonstrate a particular subgroup cohesive relationship. The common interests of a community derived from k-cores are important for topic-specific semantic social web query services to discover similar resources from this high cohesion level subgroup.

3.3 Semantic Social Web Query Services

We provide different level of semantic query services in our semantic social web portal: basic semantic query services, advanced semantic query services, and semantic social web query services:

- Basic semantic query services: The initial contribution of this article is to combine the tagging system's folksonomy with ontology to achieve basic semantic query services. This service provides people or agents to effectively access clustering of blog information through tags and related tags.
- Advanced semantic query services via ontology+tags: In this service, user enables conceptual semantic query services with relevant tags. The conceptual semantics can be defined as a channel declared from ontology with relevant tags in the tags cloud. In other words, the search space for this

service is classified and focused so the search time is reduced and accuracy is also improved.

- Semantic social web query services via SNA+ontology+tags: In a blog ontology, we define properties to describe the relationships between bloggers, tags, and articles. Additionally, the important SNA measurement attributes are also declared in a blog ontology. Therefore, we can leverage the power of SNA measures from dynamically generated relations through blogger's daily activity events to enhance this service. We propose two possible scenarios for this service that could justify our hypothesis [2]:
  1. Scenario One: I would like to search authors and their blog articles with "cuisine" tag paired with "restaurant" keyword in the associated title or content of the article collected from the entire blog community. Furthermore, please present these authors' names and their associated titles of article in a decreasing order of authors' indegree measures:

```
prefix blog: <http://blog.nccucs.org/blog.owl#>
prefix rdf: <http://www.w3.org/19999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?Author ?Article
WHERE
{?Article rdf:type blog:Article
 ?Article blog:has_articleTag blog:cuisine
 ?Article blog:has_author ?Person
 ?Person blog:person_ID ?Author
 ?Person blog:person_indegree ?Popularity
 FILTER {regex(?TitleOfArticle, "restaurant") ||
 regex(?ContentOfArticle, "restaurant"))}
  }
ORDER BY DESC (?Popularity)
```

  2. Scenario Two: I would like to search blogger names and their articles from the cuisine channel for those of whom are known by authors presented in scenario one. Furthermore, please present these blogger names and their associated titles of article in a decreasing order of authors' indegree measures:

```
prefix blog: <http://blog.nccucs.org/blog.owl#>
prefix rdf: <http://www.w3.org/19999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?Author ?Friend ?TitleOfFriendArticle
WHERE
{........
 Codes Same As Scenario One
 ........
?Person blog:has_knows ?friend
?friend blog:person_ID ?Friend
```

[2]The embedded codes are shown as SPARQL query language but users do not require to have knowledge of SPARQL syntax in order to execute semantic social web query services.

```
?FriendArticle blog:has_author ?friend
?FriendArticle blog:has_channel blog:CuisineChannel
?FriendArticle blog:article_title ?TitleOfFriendArticle
?FriendArticle blog:article_description ?ContentOfFriendArticle
FILTER {regex(?TitleOfFriendArticle, "restaurant") ||
regex(?ContentOfFriendArticle, "restaurant"))}
  }
ORDER BY DESC (?Popularity)
```

Compared with Technorati [3], it only provides limited independent search services for user from his input blog posts, tags or directory where user can not have semantic (social web) query services for any possible relevant outputs using his previous search results. So user can not search the most influential blogger friend's articles or he can not search high similarity articles from those bloggers with certain higher level of SNA indegree measures.

4 Semantic Social Web Blog Portal

In this research, a semantic social web blog portal was constructed to exploit the incentives of bridging Web 2.0 ↔ Web 3.0 where users could enjoy semantic social web query services on this portal. This portal structure is a layer schema shown as Fig. 1. In the bottom layer, crawler collects semi-structured HTML blog pages, structured RSS or FOAF context information, and free tags. Both RSS 1.0 and FOAF ontology schema are based on RDF(S) so their semantics are explicitly specified. Then, we extract and store the crawler's collected information in our local repository. In the ontology and tags annotation layer, we mash up the blog ontology and the topic ontology with collected free tags from social web annotation by folksonomy. The blog information diffusion patterns will be analyzed by using SNA software Pajek to derive important SNA measures, such as indegree, outdegree, closeness, betweenness, and k-cores, etc[20]. Finally, we provide semantic social web query services for users to satisfy his best interested.

4.1 Data Collection

WRETCH is the biggest BSP platform in Taiwan with more than 2 million registered bloggers so huge amount of living and recreation information were available for our experiment on the research issues of bridging of Web 2.0 ↔ Web 3.0. After filtering out insignificant noise data, the number of useful bloggers information samples in our analysis is around 108,518 bloggers. The period of time for our data collection was one month spanned from Sep. 09 2006 to Oct. 09 2006.

[3] http://technorati.com/

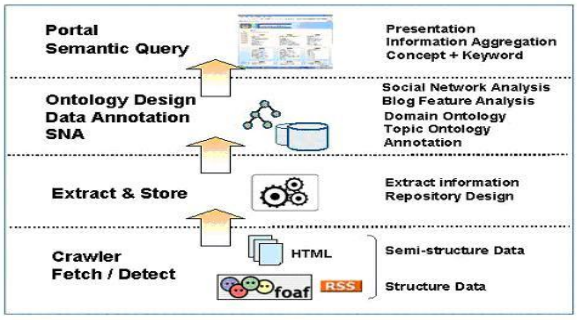

Fig. 1. A layer conceptual schema for construction a semantic social web blog portal

4.2 Data Analysis

In our mashup model, the free tags collected from users are usually 2-word or 3-word Chinese words (or characters) to annotate their daily real life's living activities. The scan and parsing processes of Chinese characters are different from the English free tags. There are no spaces between Chinese characters so we use regular expression to extract the meaningful high frequency 2-word or 3-word tags as our folksonomy final consensus social web annotations. With no surprise, the distribution for the top 300 tags is shown as power law that is similar to lots of other studies [12].

Initially the tags addressed by blogger in the *WRETCH* only imply that the taxonomy of blog articles can be classified as one of 16 broad channel categories, such as living, cuisine, music, drama, travel, etc. When we carefully examined the tags, we surprisingly found that those of significant 54,824 bloggers (approximate to 50% of 108518 bloggers) with their addressed 1046 tags were converging to some of high frequent 521 2-word and 197 3-word tags. And these tags were evenly distributed to our 16 broad channel categories. This demonstrates that the social consensus opinions are possibly formulated in terms of folksonomy tagging. We are expecting a more powerful folksonomy annotation scheme can be realized in a near future as long as we have more versatile ontology+tag structure.

4.3 Blog Ontology

The blog ontology describes the profile of a blogger with his blog articles (see Fig. 2). The profile of a blogger is very similar to FOAF that defines a blogger's personal ID, friend relationship, and mbox, etc. The attributes of each blog article include article title, date, feedback comment, and trackback, etc. In addition, the SNA index measure is defined as one of a blogger's profile attributes. Therefore, SNA analysis capabilities were embedded into blog ontology to serve our SNA+ontology+tag semantic social web query services.

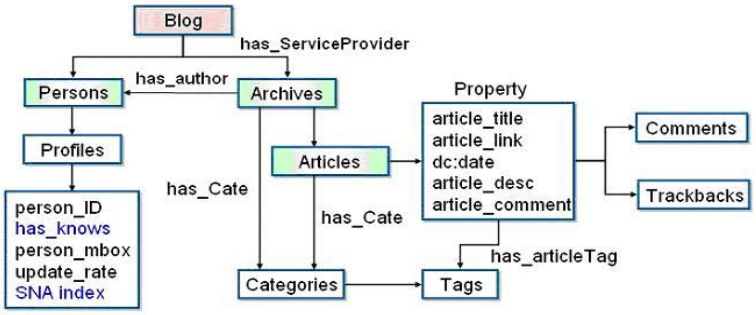

Fig. 2. The blog ontology describes the profile of a blogger with his blog articles

The blog ontology is declared as OWL ontology language, where property can be classified as two types: object property and datatype property. For example, the domain and range of the *has_author* object property are declared respectively as *Archives* class and *Persons* class, where *Archives* is the superclass of both *Articles* and *Categories* classes. Based on this object property, we describe the abstract relationships between a blogger and his blog articles. The datatype property allows us to define a concrete XML-Schema attributes, such as SNA index, for $Profiles$ subclass for further arithmetic operations.

4.4 Topic Ontology

The blog articles in the *WRETCH* were classified into one of the 16 broad topic channels based on their attachment tags. The design processes of broad classification of blog article channel will be shown as three steps (see Fig. 3): First, we subjectively declare 16 broad topic channel as instances under their superclass *Channel*. The 16 broad topic channels are life, cuisine, music, etc, where *Channel* and *Tag* are subclasses of *Category* superclass in the topic ontology. Second, a set of possible tags we consider for each channel are those with higher frequent 2-word or 3-word tags presented by users. Third, if a new blog article has attachment tags that match at least one of higher frequent tags in the set declared for one of a broad topic channels, then this new blog article will be automatically classified to that channel.

4.5 Social Web Annotation

The goal of Web 1.0 annotation is to create a well-defined and computer understandable structure knowledge base e.g., ontologies, whose content mirrors that of the WWW. The biggest challenge for bridging of Web 1.0 $\leftrightarrow$ Web 3.0 is the terms mining from the Web can not be automatically and exactly fitted into the ontology that defines the vocabularies for the target knowledge base

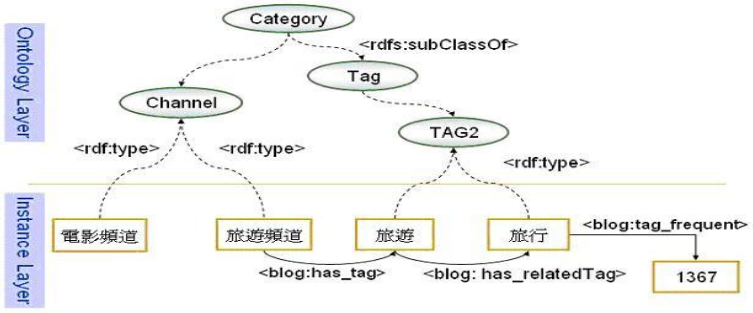

Fig. 3. Topic ontology - Channel and Tag are subclasses of Category so we can automatically mash up ontology data and search model with the folksonomy tagging system services

[7]. Therefore, most of the semi-automatic annotation systems usually apply machine learning techniques to recognize new class instances and relation instances mining from the Web. In the folksonomy annotation for bridging of Web 2.0 ↔ Web 3.0, the granularity of class instances and relation instances are restricted to the resource targets that can be clearly tagged by folksonomy. The folksonomy of social web annotations are explicitly collected from tags or implicitly initiated by users from their activity events. These explicit tags and implicit events are precise terms that describe the instances and relations corresponding to our ontology schema.

The objective and granularity of tags for describing instances and relations that corresponding to the target resources can be further refined if we have more elaborate social web annotation system in the future. As semantic wikipedia in [23], we might allow users to enable semantic tags similar to *typed links* and *attributes* two kinds of property for describing corresponding abstract relationship and concrete attributes within/between entity. Then various levels of reasoning for discovery of semantic relationship among taggers, tags, and resources can be achieved.

Our semantic social web annotation system takes three inputs either collected by web crawler or computed by local software agent. The first is HTML blog pages with hyperlinks , comments, and trackbacks context. The second is RSS context with permalink, publication data, author, and description attributes. The third is tags, channel, and SNA indices computed via agents. They are all stored in a local database and to be mashed up for afterward semantic social web query services (see Fig. 4).

4.6 Semantic Social Web Blog Portal Testbed

An online semantic social web blog portal testbed (see Fig. 5) was constructed based on previous layer conceptual schema (see Fig. 1) to experiment our

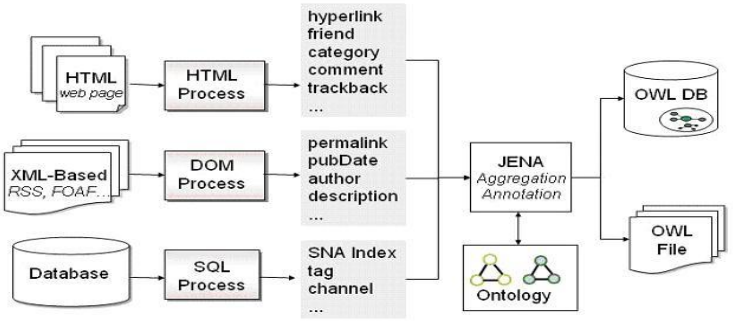

Fig. 4. Semantic social web annotation from three inputs of data sources for mashup purpose to enable semantic social web services

mashup model. The crawler collects all of the necessary context information from the *WRETCH BSP*. The context information shown in Figure 4 were processed to create relevant class and relation instances defined in the blog ontology and the topic ontology (see section 4.3 and section 4.4). This semantic social web annotations for folksonomy were automatically generated except in the bootstrapping stage where we have to analyze the blog site dependent context to specify our initial lightweight ontology schema. A variety of important SNA measures, such as indegree, closeness, betweenness, and k-core, were computed via Pajek SNA software. [4] to provide semantic social web query services shown in section 3.3.

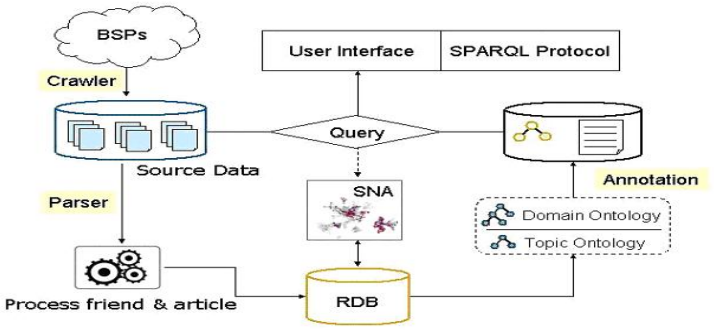

Fig. 5. The semantic social web blog portal to experiment our Web 2.0 ↔ Web 3.0 bridging model

[4] http://valado.fmf.uni-lj.si/pub/networks/pajek/

5 Conclusions

The goal of this research is to exploit the incentives of bridging Web 2.0 ↔ Web 3.0 via building a semantic social web blog portal. On the Web 2.0, we usually use tagging system to label all kinds of Internet resources. Web 2.0 is a folksonomy social web, where we effectively search what we are desirous of information through tags. The tagging system enables the wisdom of crowds and surprisingly social consensus can be derived from these voluminous and unregular tags. Contrarily, Web 3.0 (semantic web) is aiming at using ontology for effectively information search under taxonomy classification. We have justified that the concepts of folksonomy and taxonomy can be mashed up together to achieve semantic social web query services via bridging of Web 2.0 ↔ Web 3.0. That allows us to leverage search capabilities from both bottom-up folksonomy indexing and top-down taxonomy ontology two techniques.

Conceptually, tags in the tagging system are equivalent to terms mining from the WWW in the conventional annotation system. The terms mining from the Web are usually defined as instances that are related to a particular class or property in ontology. But tags from the folksonomy are usually instances related to a particular class. Therefore, all of the relation instances have to be created dynamically following the ontology schema. The relation instances that describe the relationships between bloggers, tags, and blogs, are generated from blogger's daily activity events based on our blog ontology. Although users can effectively search information by folksonomy tagging system in Web 2.0, we still have the capacity to improve search capability via social network analysis (SNA). A real SNA-based semantic social web query services could possibly encourage users to find out what they are really interested in because well-organized topic-specific ranking contents are ready for user to enjoy.

Acknowledgements

This research was partially supported by Taiwan National Science Council (NSC), Under Grant No. NSC 95-2221-E-004-001-MY3.

References

1. Ali-Hasan, N. and Adamic, L. A., Expressing Social Relationships on the Blog through Links and Comments. `http://www-personal.umich.edu/~ladamic`.
2. Berners-Lee, Tim, et al. (2001). The Semantic Web. Scientific American, May.
3. Bojars, U. , Breslin, J. G., and Moller, K. (2006). Using Semantics to Enhance the Blogging Experience. Proceedings of 3rd European Semantic Web Conference (ESWC 2006), 679-696.

4. Brooks, C. H. and Montanez, V. (2006). Improve Annotation of the Blogosphere via Autotagging and Hierarchical Clustering. WWW 2006, May 23-26, Edinburgh, Scotland.
5. Cayzer, S. (2004). Semantic Blogging: Spreading the Semantic Web Meme. XML Europe, Apr. 18-21, Amsterdam.
6. Chin, A. and Chignell, M. (2006). A Social Hypertext Model for Finding Community in Blogs. HyperText (HT'06), Aug. 22-25, Odense, Denmark.
7. Craven, M., et al. (2000). Learning to construct knowledge bases from the World Wide Web. Artificial Intelligence, 11, Elsevier, 69-113.
8. Dill, S., et al. (2003). A case for automated large-scale semantic annotation. Journal of Web Semantics, 1(1), 115V132.
9. Ding, L., et al. (2005). How the Semantic Web is Being Used: An Analysis of FOAF Documents. Proceedings of the 38th Hawaii International Conference on System Sciences.
10. Gruber, T. Ontology of Folksonomy: A Mash-Up of Apples and Oranges. `http://tombruber.org`.
11. Gruhl, D. et al. (2004). Information Diffusion Through Blogspace. WWW 2004, May 17-22, New York, USA.
12. Hotho, A. et al. (2006). Information Retrieval in Folksonomies: Search and Ranking. Proceedings of 3rd European Semantic Web Conference (ESWC 2006).
13. Karger, D. R. and Quan, D. (2004). What Would It Mean to Blog on the Semantic Web?. The Third International Semantic Web Conference(ISWC 2004), Springer-Verlag.
14. Kiryakov, A., Popov, B., and Terziev, I. (2004). Semantic Annotation, Indexing, and Retrieval. Journal of Web Semantics, 2(1), 49-79.
15. Kumar, R., Novak, J., Raghavan, P., and A. Tomkin. (2004). Structure and evolution of blogspace. Comm. of the ACM, 47(12).
16. Markoff, J. (2006). Entrepreneurs See a Web Guided by Common Sense. New York Times, Nov. 12, `http://www.nytimes.com`.
17. Marlow, C. et al. (2006). HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead, HyperText (HT'06), Aug. 22-25, Odense, Denmark.
18. Matsuo, Y., Mori, J., and Hamasaki, M. (2006). POLYPHONET: An Advanced Social Network Extraction System from the Web. WWW 2006, May 23-26, Edinburgh, Scotland.
19. Mika, P. (2005). Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks. Journal of Web Semantics, 3(2-3), 211-223.
20. Nooy, de W., Mrvar, A. and Batagelj, V. (2006). Exploratory Social Network Analysis with Pajek. Cambridge University Press.
21. O'Reilly, Tim. (2005). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. `http://www.oreillynet.com/lpt/a/6228`.
22. Uren, V., et al. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Journal of Web Semantics, 4(1), 14-28.
23. Völkel, M., et al. (2006). Semantic Wikipedia. WWW 2006, May 23-26, Edinburgh, Scotland.
24. Wasserman, S. and Faust, K. (1994). Social Network Analysis: Methods and Applications. Cambridge University Press.

Improving Flickr discovery through Wikipedias

Federico Gobbo

Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
via Mazzini 5, IT-21100, Varese, Italy
federico.gobbo@uninsubria.it

Abstract. This paper explores how to discover unexpected information in existing folksonomies (serendipity) using extensive multilingual open source repositories as the underlying knowledge base, overcoming linguistic barriers at the same time. A web application called Flickrpedia is given as a practical example, using Flickr as the folksonomy and diverse natural language Wikipedias as the knowledge base.

Key words: Flickr, Wikipedia, Multilingualism, Folksonomies, Serendipity

1 Introduction

Adding meaningful metadata to web content, in order to increase the utility of information by improve the precision of information retrieval to search engines, is one of the most desired feature by any user. Folksonomies are a tentative effort toward this goal. The term 'folksonomy' is a fusion of 'folks' and 'taxonomy' and was originally used in cognitive anthropology studies, but only very recently it became popular with a specialized meaning [9]. A folksonomy is a taxonomy made by tags or labels, usually single-word metadata attached to online items (documents, photos, videos, etc.), in order to add contextual meaning to the items themselves.

Unlike traditional taxonomies, as for example the Linnaean system used in life sciences, there is no explicit hierarchy between tags nor tags are exclusive – e.g. the photo of a cat may be tagged as 'cat' and 'european' and 'animal', but there is nothing that say that all cats are animals: tags can be seen as common facets of the item itself [6]. While in traditional taxonomies there is a central authority that controls its structure, in the case of folksonomies there is no one [5] – undoubtely this is the main reason why folksonomies are becoming more and more popular among web resource users.

Consequently, each tag has two different scopes at the same time: the user's defined one – *personimy*, [5] – and the social shared meaning – *consensus*, as the wide use of tag suggestion interfaces in web applications suggests. Social

meaning emerges when the distribution of tag use converges on some terms, and the distribution curve of tag popularity follows a 'long tail' [4, 1]: very few tags are most used (high consensus, low personimy), and a lot of tags are used once or few times by the majority of users (low consensus, high personimy).

Furthermore, consensus permits *serendipity*, i.e. users dig the web through tags finding new, unexpected and useful content, not easily accessible via traditional search engines. In fact, tags act as filters, i.e. a query on more tags returns the items tagged with any of the given tags – or with all tags, depending on the application [2]. The purpose of this paper is to improve serendipity allowing people to dig folksonomies regardless of the natural language they master.

2 Serendipity and multilingualism

Folksonomies share common problems with traditional taxonomies, due to the fact tags are words, i.e. alphabetical strings meaningful in some natural language. In particular, there is no synonym (different word strings, analogue meaning) nor homograph (identical word string, totally different meaning) control. In fact, there is no restriction to what people can write as a tag, i.e. no controlled language: people can externalize their free word association through tags, which respect their own mental models. Consequently, folksonomies lack in standardization, i.e. different strategies in tag encoding are possibles, as for instance dates (28-03-2008, '2008March3', '3r̂d March 2008' and so on) or in the case of compounds ('nice-cat', 'nice_cat', 'nicecat'), not to mention misspellings, so frequent that tag literacy education was advocated [3].

2.1 Folksonomies and the digital linguistic divide

One of the existing problems behind folksonomies not fully explored until now is multilingualism. As anedoctical evidence suggests, every tag is written in a human language and users are inclined to write in the languages they are comfortable in. It is certainly desiderable for a user not comfortable in English or other big language (in terms of presence in the web) to search and find tags using a search engine interface in his or her tongue, while the engine searches the corresponding tags in English and in other major human languages.

To do so, the user needs to specify both the tag looked for and the natural language in which it is written in a special web application, which extracts the pairs language-tags in every available language before passing the tags to the folksonomy search engine. Our claim is, when searching in 20 natural languages at same time some interesting photo will be found, that would be undiscovered through a single language search (i.e., serendipity improves).

2.2 Adding multilingualism to Flickr through Wikipedias

Flickr, a Yahoo! company, is one of the most popular online photo web applications – e.g., more than 2 million photos are found if 'flowers' are searched, at 2007, April the 11th. In Flickr, users can browse or search photos through tags, a feature that certainly contributed to its popularity. Moreover, some open source APIs are available[1] and people can make queries to the Flickr repository through an authentication key given on request. For our application, the language of choice for the API is Ruby, and the development framework is Ruby on Rails, as it is easy to produce clean code and reliable web application very quickly [7, 8].

In our prototypical web application, *Flickrpedia* (named derived from 'Flickr' and 'Wikipedia'), users can make queries in Flickr writing a tag specifying its natural language. The system crawls the Wikipedia in the corresponding language and look for an appropriate page. For example, if the user is a German-speaker and he is fond of airplanes, he may put the following pair `German:Flugzeug` and the system, which can manage case-sensivity, will look for the following page in the German Wikipedia:

```
http://de.wikipedia.org/wiki/Flugzeug
```

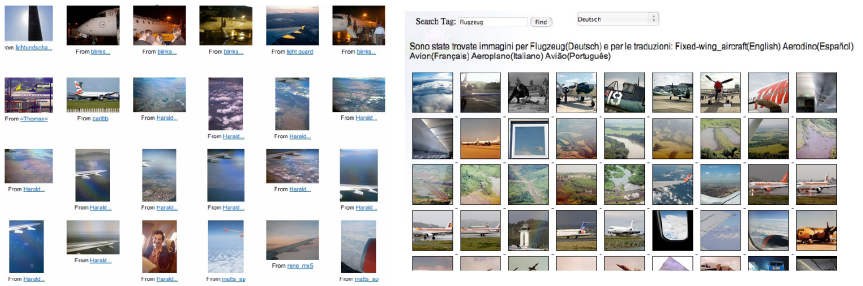

Fig. 1. The same query on Flick (on the left) and Flickrpedia (on the right).

where `de` is the natural language ISO code and `Flugzeug` indicates the corresponding web page. With the help of regular expressions, Flickrpedia parses the web page and extracts the existing language pairs of the same topic (airplanes) in other languages from the appropriate web page box known as "in other languages", e.g. `English:Airplane`, `French:Avion` – but also minority languages, as `Basque:Hegazkin` (see Fig. 1). The topic names are passed to Flickr as search queries and thumbnails are given to the user.

While Flickr finds less than 10,000 photos (2007, April the 11th) for the tag 'flugzeug' Flickrpedia finds more than 20,000 for the same query, giving a lot of unexpected and relevant photos.

[1] See `http://www.flickr.com/services/api`.

3 Conclusions and further directions

This paper has shown that serendipity in Flickr can be improved through the exploitation of Wikipedias's URLs as translation sources. The main advantage is that Flickrpedia should only store the wikipedias according to the existing natural languages – actually, 85. This approach wants to suggest that large and extemporaneus shared information repositories, like Flickr, can be managed through other semi-structured information repositories as the wikipedias – as known, wikipedias are the result of a wide and magmatic community of contributors, even anonymous. Moreover, Flickrpedia, if refined out of its actual prototypical phase, may help users with poor knowledge of major languages to retrieve information only through their lesser-used languages.

Flickrpedia is far from perfect: homographies are still unmanaged, even if wikipedias have disambiguating pages, and it is not clear which wikipedias to choose in order to optimize serendipity. By the moment, the parsed wikipedias are the biggest ones in terms of wiki pages, but this doesn't give any guarantee of serendipity augmentation. Finally, the API given by Flickr is a severe limit: up to 20 tags can be inserted in a single query request, and up to 60 thumbnails may be given.

However, this approach isn't limited to Flickr as the underlying folksonomy. Our research direction is towards generalization, i.e. users can choose the appropriate folksonomy performing multilingual queries. Finally, specific and precise metrics for serendipity are needed, in order to achieve more formally sound results.

References

1. C. Anderson. *The Long Tail*. The Random House Group, 2006.
2. S.A. Golder and B.A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2), December 2006.
3. M. Guy and E. Tonkin. Folksonomies: Tidying up tags? *D-Lib*, 12(1), January 2006.
4. A. Mathes. Folksonomies: Cooperative classification and communication through shared metadata. December 2004.
5. E. Quintarelli. Folksonomies: power to the people. June 2005. ISKO Italy-UniMIB meeting.
6. Patrick Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006*, May 2006. Edinburgh, Scotland.
7. D. Thomas. *Programming Ruby*. The Pragmatic Programmers, 2005.
8. D. Thomas and D. Heinemeier Hansson. *Agile Web Development with Rails*. The Pragmatic Programmers, 2005.
9. T. Vander Wal. Explaining and showing broad and narrow folksonomies. February 2005.

Learning by tagging: The role of social tagging in group knowledge formation[1]

Jude Yew[1], Faison P. Gibson[2], Stephanie Teasley[1]

School of Information North, University of Michigan, 1075 Beal Ave., Ann Arbor, MI 48109[1]

College of Business, Eastern Michigan University, Ypsilanti, MI 48197[2]

Abstract. This research presents a case study on the use of Social Tagging in an undergraduate classroom at the University of Michigan during the Fall 2005 semester. Students were between 20 and 22 years of age. Students tagged their individual blog posts to contribute to themes and conversations in an online learning environment. Using content analysis of the blog posts and tags as well as semi-structured interviews, the study examines the role of online social tagging for tracking and aiding group knowledge formation.

Introduction

This paper presents a case study from an ongoing research project that investigates knowledge and community formation in online learning environments that employ social tagging. These learning environments allow the user to organize and display online content, such as blogposts and

[1] *This work has also been published under the Creative Commons license at the MERLOT Journal of Online Learning and Teaching (JOLT) at the following URL: http://jolt.merlot.org/vol2no4/yew.htm*

bookmarks, with meaningful keywords or tags presented in a public and collaborative manner. Such labeling of online content potentially allows the individual learner and the community to use technology and social conventions to organize knowledge, coordinate with others, and facilitates the sensemaking efforts of the community (Mathes, 2004).

This study makes the argument that social tagging systems employed within a learning community can both facilitate the process and provide evidence of knowledge formation within the group. To investigate this, we first put forward a theoretical argument for why social tagging systems should be employed to facilitate the production of group knowledge. We then present an analysis of an undergraduate business school class' online learning environment that utilized social tagging.

The case for social tagging

Tagging describes the activity of marking online content with keywords, called "tags", as a way to organize content for future navigation, filtering or search. Tags are not based on a controlled vocabulary, but rather are left to the user's wishes, although as shown in this study group norms and social processes can play a significant role in an individual's choice of tags leading to fairly consistent assignment of specific tags (Mathes, 2004). This act of assigning tags to categorize an object is an act of knowledge production as it makes apparent the mental models, or internal representations of knowledge, that one uses to associate with the object (Pauen, 2002). The argument being made here is that allowing students to associate keywords to objects we are enacting the associative structure of knowledge formation (von Anh & Dabbish, 2004). New knowledge is formed in the allocation of tags, as the individual has to make sense of the new object by associating it with prior understandings and classification of objects. For instance, by categorizing a digital photograph with the tag 'vacation', we are immediately providing information about the content of the photograph without actually having to view it. Also, the tag "vacation" provides information to others about how we have contextualized the photo. Thus, the use of tags can function both as a way to facilitate the formation of new knowledge as well as to provide evidence of how this knowledge evolves over time.

Tagging is social because the tags are visible to the whole group with the potential for influencing the tags adopted by each group member. We

believe that social tagging systems employed within a learning community can facilitate knowledge formation within the group. In addition, social tagging can provide evidence of knowledge formation to both the group members and to researchers/analysts. In a class, the tags used by individual students to categorize online content also functioned as a "repository" of how that particular student made sense of and assimilated the material being taught in the class (Argote, 1999; Weick, Sutcliffe & Obstfield, 2005). When tags are made public and shared, other students in the class are able to tap into the knowledge being formed by the individual student. Students are able to view the tags used by others and employ those tags to inform their own understanding, creating an iterative learning loop (Russell, Stefik, Pirolli & Card, 1993). Additionally, the tags employed by one member of the class can "self-propagate" and become a "linguistic meme" that enables the entire class to organize and coordinate their online discussion, and in the process of doing so, establishes a common understanding of the material being taught (Heath & Seidel, undated).

Methodology

The setting

This study took place in Business Information Technology 320 (BIT320), a database and Information class offered at the University of Michigan. The class was offered to undergraduates aged 20 to 22 at the Business school and a large part of the class was devoted to group work where students were expected to create information databases based on the technologies taught within the syllabus. BIT320 also used blogs and RSS (an XML format for syndicating blog content) to create an online space where both the professor and the students could share their knowledge. The class website was dubbed the "Class Remix" to encourage participants to improve upon, change, integrate, or otherwise "remix" the group's knowledge contributions similar to Lessig's notions of a remix culture (Koman, 2005). Participation in the Class Remix was mandated through a class policy that stipulated 5 blogposts per week that were then aggregated in the site (Here on the web and pictured in Fig. 1). Students were encouraged to create a vibrant learning community where group knowledge was built collectively by sharing relevant links, questions, answers, and observations of the material taught in the class.

In this environment, students could post about new ideas, or they could effectively respond to the contributions of others by writing a response in their own blog and linking back to the original poster. In this way, conversations (initial post, comment, response to comment, etc.) effectively occurred across student blogs. When engaging in these sorts of conversations, students were encouraged to reuse at least some of the tags that previous posters had used, as well as, adding any new tags they might find relevant. In this way, whole conversations came to be grouped by tag and were made findable by tag. A limitation of the system was that once a post was tagged and saved, the tags could not be changed.

Fig. 1: Screen capture of class "remix" website (04/14/06)

Unlike more orthodox and prescribed forms of classification, social tagging allowed the users in the community to assign any keyword/category to their contribution that they deemed relevant. Various visualizations, such as the use of tag clouds on the class website (highlighted in blue lower right corner of Figure 1), helped members of the class to be aware of the current and most frequently submitted topics/posts. The class remix website can be seen as an archive of the students contributions, and can be used to document the students' evolving understanding and knowledge formation that has taken place during the course.

Data and methodology

Data for this study were composed of participants' contributions to the class remix website and in-person interviews. To better understand the role of the remix site in the particpants' learning, content analysis was performed on the student blog posts and the tags they employed to describe these posts. Additionally, the students' grades in the class and semi-structured interviews with seven out of the eleven participants in the class provided complementary data. In the following section, the server log analysis, the key findings generated by the interviews, and the content analysis of the blogposts are reported.

Findings

Table 1 outlines the total number of blogposts made by each student in the class during the term, the total number of tags that they associated with their blogposts and the average number of tags per blog post contributed to the class website.

The majority of the students adhered to the instructor's requirements that they contribute five blogposts a week to the class website. With the exception of three students, everyone in the class met the minimum requirements of 5 blogposts a week that was stipulated by the instructor (highlighted in Table 1 by the red line).

Table 1: Total blog posts and tags and avg. tags per post (13 weeks x 5 blog posts/week = 65 minimum required posts).

Source	Total Posts	Total Tags	Avg. Tags/Post
The Blogstar	36	75	2.0833
Musings of William h	41	72	1.7561
Matt's Musings	61	156	2.5574
jb's blog	65	150	2.3077
zee124	66	124	1.8788
Shady Waters	66	219	3.3182
Supriya	66	146	2.2121
Pink Footsie	68	154	2.2647
Tigerlily's Blog	69	119	1.7246

Kevin's Blog	70	137	1.9571
SuperMatt	72	230	3.1944
Blogonautic Solutions (instructor)	74	198	2.6757

The instructor's purpose for stipulating a minimum requirement of contributions was to encourage the students to fully utilize the system, and to ensure sustained participation from the students. The instructor's rationale for mandating participation online is illustrated in the following quote:

"… This is one of those things where initially people have some hesitation … I mean there's just all that group anxiety that comes into play and so you got to get over that hump, you got to get over it early and just start making it happen. It's also practice (that) makes it better …" (Inst1interview, 0:32:50)

As shown by the Average Tags/Post column in the Table 1, participants tended to use more than one tag to describe the content of each blog contribution, a common practice in this type of system (Kroski, 2005). Because of the great number of tags being employed, one issue that emerges is that of the *vocabulary problem* (Furnas, Landauer, Gomez & Dumais, 1987). This problem highlights the issue that there are multiple ways to describe an object/idea and that random pairs of people label an object similarly at most 20% of the time (Furnas et al, 1987). Because of the vocabulary problem, participants in the class are forced to determine exactly what should be the common vocabulary for describing their blog posts. One student described how the group made sense of multiple tags as follows:

So when you have hundreds of tags, it's really the case that only a few of them are important. And that was the case here. And so people were able to figure that out, and that we had sort of themes. So at any given point in time, maybe 10 tags would be important. (Stud2 interview, 0:13:51)

This pattern was reflected in the analysis of the server logs. In total 143 distinct tags were used 1780 times during the term. However not all tags were used equally. As indicated by the quote from Student 2 above, there were a small number of keywords that were used more frequently than others. Figure 2 highlights the 'Long Tail', or the exponential distribution, phenomena (Anderson, 2004) where a large proportion of the 143 keywords contributed were used only once or twice.

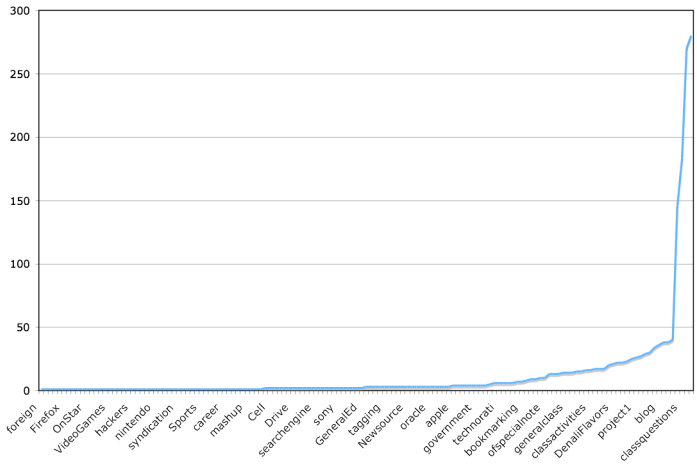

Fig.2: Tag frequency distribution .

Of the 20 most frequently used tags shown in Table 2, the top four tags (highlighted in Table 2 below) were used at least three times more frequently the others.

Table 2: Top 20 distinct tags by frequency used

Tag/ Keyword	Frequency
Technology	280
Opinionslug	270
Classquestions	183
Blogging	145
Microsoft	40
XML	38
Internet	38
Blog	36
Remixing	34
Project2	30
Databases	29
NewInventions	27
Project1	26
WordPress	25
Google	23

SQL	22
ClassIssues	22
DenaliFlavours	21
Ipod	20
Normalization	17
Weblogs	17

By investigating the timing of when certain tags were adopted and their patterns of use, the formation of group knowledge and convention can be represented. As shown in Table 3, the top four tags were adopted by the students early on in the semester and their continual use resulted in them becoming conventions for the students in the class to talk about specific subjects in their blog contributions.

Table 3: Top 4 tags by source and earliest data published

Tag	Source	Earliest date published
Technology	Kevin's blog	09-11-2005
Opinionslug	Pink Footsie	09-14-2005
Blogging	jb's blog	09-14-2005
Classquestions	Tigerlily's blog	09-15-2005

Other more specific tags like SQL, XML and Databases were used only during the part of the term where that subject was the most heavily discussed in class. The instructor of the class represented the phenomenon as follows,

"… a tag winds out being a term or label that people introduce. They introduced it to have a shorthand for referring to some phenomenon. And then if they re- use this term at given points in time, they're saying that phenomenon is there. And so what winds up happening is you see that there are themes, and basically these are recurring uses of tags." (Inst1 interview, 0:15:47)

The formation of "themes" within the class suggests how social tagging aids with the formation of group knowledge around specific course content. The frequency of use of the top four tags and the instructor's comments support the claim that those tags are functioning as artifacts/repositories of the shared understanding between the individuals in the class (Argote, 1999). And because these tags have been used by every member of the class at one point or another during the term, group knowl-

edge or shared understanding has been formed as a result of the "learning loop" that occurs through their use (Russell et al, 1993).

The differential use of tags

Content coding of the student interviews revealed that not all tags were used in the same way. There were two kinds of tags; functional tags (e.g. "opinionslug" or "classquestions") and content tags (e.g. "technology" and "XML").

Functional tags are labels that indicate some form of utility or function to the members of the class. For example, the "classquestions" tag was deliberately used by the instructor of the class as a way to easily indicate and highlight questions or problems that the students may be having with the material being taught. One functional tag, "opinionslug", was a keyword first coined by a student, Pink Footsie. "Opinionslug" was used to indicate contributions that were personal opinions or views of both the content matter or administrative aspects of the class. According to Student 2,

> "… at first it was only Pink Footsie who used that ... cause she was the one who invented it ... but then as we started reading more and understanding what she meant by 'opinionslug' ... we definitely all started using it ... but if you just started looking at this (tag) you would probably have no idea what it was ... So it was a kind of inner group understanding." (Stud2 Interview, 0:27:58)

From the illustration of the use of the "opinionslug" tag, we can see that an explicit purpose/function is signaled through its use and it prepares the reader of the contribution to both understand and react appropriately to what is being said in the blogpost.

Another example of a functional tag is "classquestions" which seemed to be a term coined by Tigerlily's blog but was actually stipulated by the instructor to create threads of interaction that could be retrieved by the students later on. Student 2 indicated that,

> "he(the instructor) told us that if ever we had a class question we had to call it "classquestion" ... and if you actually clicked on classquestions you would actually see a stream." (Stud2 interview, 0:33:48)

The adoption of tags to continue a thread of interaction was practiced by Student 2, who explained that the popularity of certain tags had to do with

the fact that they highlighted interesting threads of conversation:

"It definitely had to do with the fact that she (a classmate) would have had to have an interesting enough post where I would reply to it or I would make a post about her post ... and so then when I was picking out my tags I would look at what she called it ...just because I am conscious of that and want to make sure that you could find out stream of conversation ... if it was something really boring that no one answered then it probably wouldn't catch on." (Stud2, 0:29:26)

Thus we can see that functional tags like "opinionslug" and "classquestions" signaled an explicit purpose and their high frequency of use points to the fact that the convention of using these tags to highlight the function of a blog post became a social norm with in the class.

In contrast, content tags were topics that the class dealt with explicitly. There was a certain amount of ambiguity in how content tags were used and perceived by the students in the class. This ambiguity could be because content tags embodied meanings that went beyond the shared understandings of the students and have significance outside of the class as well. An example of a content tag and how it is used can be seen in the Student 1's comparison of how her use of the "XML" tag differs from the "opinionslug" tag:

"Well with XML it's harder ... if I had a question about XML and someone answered it and put XML in the tags... it's fine but there's so many different things to call it ... you know it could have been about databases, it could have been about writing code ... whereas with "opinionslug" it was very obvious you were going to call opinionslug because you were basically preaching on your opinions." (Stud1 interview, 0:30:40)

This sentiment was shared by Student 3, who used the content tag "technology" in the following way;

"For example, when I first started my blog, I was trying to come up with a common thread to a lot of the things, so I use the word "Technology" a lot in my blog. That's such a vague word you know ... And at the same time if I was just looking, or had a couple of minutes to spend, then I would say, "give me something interesting about 'technology' that's going on" and I wanted that broad topic." (Stud3 interview, 0:26:30)

What is highlighted from the student quotes, is the issue of polysemy, or the multiple meanings of words (Furnas et al., 1987). Polysemy is a double-edged sword in the use of social tagging systems. It would seem that

the use of popular content tags like “technology” were deliberately used to signal the content of the blog post and appeal broadly to as many individuals as possible. However the problem with such tags is that they are also highly ambiguous and often have to be paired up with other terms such as “ipod” and “Microsoft” to qualify their meaning. As highlighted by its near-ubiquitous use in the class’ learning environment, many of the blog posts that had anything remotely connected to the class would use the keyword “technology”. However, because of the ambiguity of the term “technology”, multiple tags were used to qualify meaning. As a result, many tags associated with blog posts tended to be used only once or twice and fall from use after a while. This pattern, as highlighted in Figure 2, explains the existence of the long tail of keywords where a large proportion of the 143 unique keywords contributed were used only once or twice and then was relegated to a low and minute position in the tag cloud.

From the analysis of how tags are used by the students, we can see that it is much more difficult to base assertions of group knowledge formation around popular or frequently used tags. What is shown is that the students used tags according to a shared notion of the tags’ function. Very often, tags were used to continue threads of conversation and to signal the content of the blogpost. As a result, the group knowledge that is formed around the students’ use of tags does not necessarily represent their understanding of the content but rather the shared understanding of how the tags are used to signal norms of participation within the class.

To further explore how tags were used, content analysis of the text in the students’ blog posts was conducted to determine the correlation of ideas and concepts in the text of the students’ blog posts with the tags that were used. However, it is obvious from the previous section that keywords like “technology” were broad and that the content analysis of the students’ blog post would not necessarily reveal any correlation between the content of the students’ contribution with the keywords chosen.

For example, one particular blog post contributed by Matt’s Musings was labeled with the following tags; “opinionslug”, “technology”, and “blogging”. Content analysis of the text in the blog post produced a word frequency analysis that highlighted only one co-occurrence of the tags used with the content of the post. The tag “technology” was a word that was appeared once in the textual content of the blog post. The subject of the blog post was mainly about cellular phone technology between the US and other countries. So in general the “Technology” tag only represented

the post very broadly. What is interesting to note is that functional tags such as “opinionslug” tend not to co-occur in the body of the post as they represent the function, not the content of the post. Again this highlights the differentiation between the purpose and use of content versus functional tags.

The idea of a shared vocabulary is crucial to the formation of group knowledge. Having a common language enables the processes of establishing mutual beliefs and mutual assumptions in group communication, processes that are essential to the formation of a community (Clark & Brennan, 1991). As had been indicated in the previous section, tags like “opinionslug” and “classquestions” functioned as a way for the students to communicate and interact with each other. It was a way for them to signal the intentions of their contribution and to publicly solicit and provide help to each other. Student 3 articulates this sentiment in the following comment;

> “On the occasions when I answered questions, which was rare, or when I responded to somebody else's blog, I tried to use the same tags that they (the other students) used when they wrote ... I would intentionally try and incorporate those into my tags, and maybe if it had to do with something else, also include the other tags just to try to cover my bases so that somebody else could follow the same kind of logic or thread-line, get to their blog and then my blog.” (Stud3 interview, 0:21:08)

Thus, the tags proved useful to learning because they provided a common vocabulary with which the students are able to interact with each other. This aspect of interaction seemed to be the predominant learning benefit that the students experienced during the term.

It was these interactions, made public on the class “remix” website through the tags, that the students valued. For them, the system added a new layer of social interactions on top of the physical interactions that were going on during the class. Student 2 makes this point as follows:

> “I think that this contributed to the class so much ... you know it made us more friendly with each other ... we'd come in the next day and we'd be like "Oh my god! Did you read what Student x wrote." Literally, it was so nerdy but we did. And ... the professor would start cracking jokes like "Student Y mis-spelled this word in her blog" and he would mispronounce it during lecture on purpose ... and we all got the joke cause we all read the blog. It really contributed to the bonding and how we got along with each other.” (Stud2, 0:45:26)

The role of blogging in learning

While the focus of this study concentrated on the use of social tagging, an important premise made was that group blogging might help students learn. One way to explore this premise is to test the extent to which blogging performance was correlated with performance in other aspects of the class. Fortunately, the case study provides data to perform this test. As part of the grading process, the instructor computed a blog index for each student (Table 4). This index consisted of the instructor's rating of the quality of each student's overall blog output multiplied by the total number of posts the student produced. Quality was a function of the length and relevance of student posts. This index showed a significant correlation ($r(9) = .663$, $p < .05$) between the blog index and the students' final grades less the blogging component of the course. Examining the components of the blogging index reveal that total posts is significantly correlated with the grade in other components of the course ($r(9) = .692$, $p < .05$). However, the quality of posts is not significantly correlated with the students' final grade ($r(9) = .383$, $p > .05$). These correlations suggest that students who interacted more often, by posting blog contributions to the learning remix website, tended to achieve better performance.

Table 4: Class performance with blog index and final grad

Total Posts	Post Quality	Blog Index (Total posts * Post quality)	Final Grade less Blogging Component
72	1.75	126	63
68	1.5	102	63
66	1.5	99	57
61	1.5	91.5	56
72	1.25	90	60
66	1.25	82.5	57
65	1.25	81.25	58
69	1	69	55
66	1	66	63
36	1.25	45	53
41	1	41	54

The reasons for improved performance may be varied. For one, these measures may all simply be correlated with underlying traits of the learner

such as diligence and intelligence. However, learning in higher education is by its nature an intensely social process. People communicate and process information interactively. The blogging environment, along with the use of social tagging, provided students with an environment that offered greater opportunities to interact regarding class material than could be afforded during the allotted class time. Those who took advantage of this opportunity more often performed better in other aspects of the class.

Discussion

The main hypothesis of this study is that the use of social tagging can aid with group knowledge formation in the classroom. The findings indicate that social tagging enabled the *process* of group knowledge formation as well as the labeling of that content. Social tagging enabled the students in the class to not only interact with each other through a shared vocabulary, but also develop a set of common norms and practices. For instance, the use of functional tags provided members of the class with a means to indicate the purpose of their blogposts. Blogposts tagged with "opinionslug" highlighted that the author would be getting on his personal soapbox and airing his views. This enabled other students to make a choice of either avoiding or reading that particular posting, without the need to look at the title or the body of the blogpost. Additionally, the use of the tags was a way students kept track of their interactions with each other. The class norm of using the same tags as the post that one is responding to enabled students to identify and track the interactions they had with each other.

Thus the evidence presented by this analysis strongly shows that, through the use of social tagging, the students built shared vocabulary and norms for interacting with each other in the online learning environment. This can be understood as the mechanism by which group knowledge can begin to form. Instead of uncovering the "what" of group knowledge (its content), this study uncovered instead, the "how" (its process).

References

Anderson, Chris. "The Long Tail." Wired, 12.10 October 2004. Retrieved on Oct. 13th, 2005 from http://www.wired.com/wired/archive/12.10/tail.html.

Argote, L. (1999). “Organizational Learning: Creating, Retaining, and Transferring Knowledge”. In, Organizational Memory. Kluwer Academic Publishers, pp. 67-97.

Clark, H. H., & Brennan, S. E. (1991). Grounding in Communication. In Resnick, L. B., Levine, J. M., & Teasley, S. D. (Eds.) *Perspectives on Socially Shared Cognition* (pp. 127-149), Washington, DC: American Psychological Association.

Furnas, G. W., Landauer, T. K., Gomez, L. M., Dumais, S. T., (1987) "The vocabulary problem in human-system communication." *Communications of the Association for Computing Machinery*, 30 (11), Nov 1987: 964-971.

Heath, Chip and Victor Seidel. (Undated) Language as a coordinating mechanism: How linguistic memes help direct appropriate action. Working paper, http://www.si.umich.edu/ICOS/Linguisticmemes4.2.pdf

Koman, R. (2005). *Remixing Culture: An Interview with Lawrence Lessig*. Retrieved October 19th, 2005, from http://www.oreillynet.com/pub/a/policy/2005/02/24/lessig.html.

Kroski, E. (2005). The Hive Mind: Folksonomies and User-Based Tagging. Infotangle, December 7th, 2005. Retrieved on Jan. 2nd 2006 from http://infotangle.blogsome.com/2005/12/07/the-hive-mind-folksonomies-and-user-based-tagging/

Mathes, Adam (2004). Folksonomies - Cooperative Classification and Communication Through Shared Metadata, December, 2004. Retrieved on Dec. 1, 2006 from http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html

Pauen, S. (2002). "Biobehavioral Development, Perception, and Action: Evidence for Knowledge–Based Category Discrimination in Infancy". In Child Development Volume 73 Issue 4 (July/August 2002). Retreived on 16th December 2005 from http://www.blackwell-synergy.com/links/doi/10.1111/1467-8624.00454/abs/

Russell, D. M., Stefik, M. J., Pirolli, P., Card, S. K. (1993) "Cost structure of sensemaking" Proceedings of the Conference on Human Factors in Computing Systems - INTERACT '93 and CHI '93. ACM, New York, NY, USA: 269-276.

von Ahn, L. and L. Dabbish (2004). Labeling Images with a Computer Game. In, Proceedings of ACM CHI 2004, pp. 319-326.

Weick, K., Sutcliffe, K. & Obstfield, D. (2005) Organizing and the Process of Sensemaking. *Organizational Science*, Vol. 16, No. 4, July – August 2005, pp. 409-421.

Author Index

Marco **Benini**, 21
Dipartimento di Informatica e Comunicazione,
Università degli Studi dell'Insubria,
via Mazzini 5,
IT-21100, Varese, Italy
marco.benini@uninsubria.it

Giacomo **Fiumara**, 1, 10
Dipartimento di Fisica,
Università degli Studi di Messina,
Salita Sperone 31,
I-98166 Messina, Italy
giacomo.fiumara@unime.it

Faison P. **Gibson**, 48
College of Business,
Eastern Michigan University,
Ypsilanti, MI 48197

Federico **Gobbo**, 21, 44
Dipartimento di Informatica e Comunicazione,
Universit`a degli Studi dell'Insubria,
via Mazzini 5,
IT-21100, Varese, Italy
federico.gobbo@uninsubria.it

Yuh-Jong **Hu**, 31
Emerging Network Technology Lab.
Dept. of Computer Science,
National Chengchi University,
Taipei, Taiwan, 11605,
hu@cs.nccu.edu.tw

Mario **La Rosa**, 10
Dipartimento di Fisica,
Universit`a degli Studi di Messina,
Salita Sperone 31,
I-98166 Messina, Italy

Tommaso **Pimpo**, 10
Dipartimento di Fisica,
Universit`a degli Studi di Messina,
Salita Sperone 31,
I-98166 Messina, Italy

Stephanie **Teasley**, 48
School of Information North,
University of Michigan,
1075 Beal Ave.,
Ann Arbor, MI 48109

Jude **Yew**, 48
School of Information North,
University of Michigan,
1075 Beal Ave.,
Ann Arbor, MI 48109

Cheng-Yuan **Yu**, 31
Emerging Network Technology Lab,
Dept. of Computer Science,
National Chengchi University,
Taipei, Taiwan, 11605,
g9302@cs.nccu.edu.tw